Lucas da Silva Maciel

A NOVEL SWARM-BASED ALGORITHM FOR PHASE UNWRAPPING

Dissertation submitted to the Mechanical Engineering Graduate Program of the Federal University of Santa Catarina in partial fulfillment of the requirements for the degree of Master of Mechanical Engineering.

Supervisor: Prof. Dr. Armando Albertazzi Gonçalves Jr.

Florianópolis
2014

Ficha de identificação da obra elaborada pelo autor
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

da Silva Maciel, Lucas
A Novel Swarm-Based Algorithm for Phase Unwrapping /
Lucas da Silva Maciel ; orientador, Armando Albertazzi Gonçalves Jr – Florianópolis, SC, 2014
126 p.

Dissertação (mestrado) – Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Engenharia Mecânica.

Inclui referências

1. Engenharia Mecânica. 2. Processamento de imagens. 3. Phase unwrapping. 4. Swarm Intelligence. I. Albertazzi Gonçalves Jr, Armando. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia Mecânica. III. Título.

Lucas da Silva Maciel

A NOVEL SWARM-BASED ALGORITHM FOR PHASE UNWRAPPING

This Dissertation was considered adequate to obtain the degree of Master of Mechanical Engineering and was approved on its final version by the Mechanical Engineering Graduate Program.

Florianópolis, 17 November 2014

_____
Prof. Armando Albertazzi Gonçalves Jr., Dr. Eng.
Department Chair

Examining board:

_____
Prof. Armando Albertazzi Gonçalves Jr., Dr. Eng.
Supervisor
Federal University of Santa Catarina

_____
Prof. Marcelo Ricardo Stemmer, Dr.-Ing.
Federal University of Santa Catarina

_____
Prof. Tiago Loureiro Figaro da Costa Pinto, Dr. Eng.
Federal University of Santa Catarina

_____
Prof. João Carlos Espíndola Ferreira , PhD
Federal University of Santa Catarina

_____
Analucia Vieira Fantin, Dra. Eng.
Labmetro - Federal University of Santa Catarina

This thesis is dedicated to my parents.

ACKNOWLEDGMENTS

First and above all, I thank God for his unfailing love and mercy. Thank you, Lord, for this opportunity and for never ceasing to guide me in your will.

Karina, my love, I thank you for all your patience, encouragement and endless love. It has been an immense pleasure and privilege to walk this journey by your side.

To my parents, Natanael and Walery, I am grateful for the ever-loving guidance and support throughout this journey since my earliest childhood dreams.

I would like to thank my supervisor, Prof. Armando, for believing in my crazy ideas and continually supporting my research with great insight and careful guidance.

I would also like to thank the great support given by the staff at LABMETRO, especially Rosana, who was always ready to help in every little detail.

In addition, I would like to thank the Brazilian Institute of Oil, Gas and Biofuels (IBP) for the scholarship that made this research possible.

Finally yet importantly, I would like to thank my closest friends and family for their encouragement and friendship.

“Go to the ant, you sluggard; consider its ways and be wise!”

Proverbs 6:6

ABSTRACT

The proper functioning of underground oil and gas pipelines depend on the frequent and correct monitoring of stress states. Recent developments on residual stress measurement techniques have employed optical methods allied with stress relief in order to assess the underlying stress field. These optical methods require a phase unwrapping step to interpret the acquired data correctly. Phase unwrapping has posed a challenge for many optical metrology applications for decades and saw the development of many different solutions. For the past decades, the field of Swarm Intelligence, based on the behavior observed among ants, bees and other social insects, has been studied and many algorithms have been designed to perform a variety of computational tasks. Swarm Intelligence is commonly regarded as robust and fast, which are desirable features in a phase unwrapping algorithm. This work proposes a novel approach to phase unwrapping based on Swarm Intelligence, assessing its applicability, comparing it to existing methods and evaluating its potential to future developments. The proposed algorithm is thoroughly explained and the results for several different images are presented. These results show a great potential of the proposed method, performing better than some established techniques in specific situations. This potential is assessed and suggestion for future advancements are given.

Keywords: Phase unwrapping. Swarm Intelligence. Optical metrology. Fringe processing. Interferometry.

RESUMO

O correto funcionamento de tubulações subterrâneas para o transporte de gás e petróleo depende de um monitoramento frequente e correto dos estados de tensões. Avanços recentes na medição de tensões residuais têm aplicado métodos ópticos em conjunto com o alívio de tensões de maneira a avaliar o campo de tensões no componente. Estes métodos requerem uma etapa de remoção do salto de fase para interpretar corretamente os dados adquiridos. Esta remoção do salto de fase tem sido um desafio para diversas aplicações metrológicas. Este trabalho tem por objetivo propor uma abordagem original para a solução deste problema. Neste trabalho é apresentado o algoritmo proposto assim como diversos resultados com diferentes imagens comparados com métodos consagrados.

A luz, comportando-se como onda, obedece ao princípio de superposição que por sua vez dá lugar ao fenômeno de interferência. Este fenômeno pode ser utilizado de diversas maneiras para a medição de superfícies e formas geométricas. No entanto, várias dessas aplicações, como interferometria speckle e shearografia, fornecem os valores de interesse restringidos a um intervalo de $-\pi$ a π. Assim, faz-se necessária uma operação para retomar os valores reais que produziram o resultado obtido. Esta operação é chamada de remoção do salto de fase.

Por décadas tem-se estudado diversas técnicas para realizar a remoção do salto de fase. Elas podem ser divididas em duas categorias principais: métodos que seguem caminhos e métodos independente de caminhos. Métodos que seguem caminhos aplicam uma simples equação de comparação e adição de múltiplos de 2π por toda a imagem. Elas diferem nos caminhos de pixels escolhidos. Para que o resultado seja confiável, é necessário que esse caminho evite pixels de baixa qualidade ou corrompidos. As técnicas de *branch-cut* identificam esses pixels através da teoria de resíduos e conectando resíduos de sinais opostos, ela é capaz de traçar caminhos confiáveis para a remoção do salto de fase. Técnicas baseadas em qualidade atribuem notas relativas a diferentes critérios de qualidade para cada pixel, excluindo da análise aqueles que se encontram abaixo de um limiar arbitrário.

Técnicas independentes de caminhos, como os métodos de norma mínima, assemelham-se a métodos de otimização. Estes são iterativos e procuram por um mínimo na diferença entre as derivadas da solução proposta e as derivadas da imagem original. Estes métodos são considerados bastante robustos e confiáveis. No entanto, estes também

demandam maior tempo de processamento para encontrar a resposta correta.

Em paralelo aos desenvolvimentos na área de remoção do salto de fase, cientistas têm desenvolvido técnicas computacionais baseadas no comportamento de animais sociais. O campo de Inteligência de Enxame é inspirado por insetos como formigas, abelhas e cupins e outros animais como peixes e pássaros. Estes animais têm em comum o fato de criarem sistemas organizados embora compostos de elementos simples e a ausência de uma liderança clara. O comportamento de formigas e abelhas na busca por comida e os movimentos em grupo de peixes e pássaros são os exemplos mais claros do conceito de comportamento emergente: um comportamento que, embora não explícito na descrição de seus elementos individuais, surge com a interação entre diversos desses elementos. Este comportamento emergente pode ser explicado em termos de agentes simples e independentes, regras simples e um comportamento descentralizado.

Este fenômeno tem inspirado as ciências da computação por décadas. Diversas soluções computacionais para problemas matemáticos ou operacionais têm sido propostas a partir das soluções elegantes encontradas na natureza. Exemplos dessas soluções são os algoritmos de otimização baseados no comportamento de formigas e abelhas. No entanto, pouco deste conceito tem sido aplicado na área de processamento de imagem. Quanto ao problema de remoção do salto de fase, mais especificamente, não foi encontrado nenhum trabalho que propusesse uma solução baseada em Inteligência de Enxame.

Assim, o presente trabalho propõe uma solução baseada nestes conceitos. Por causa da natureza imprevisível do comportamento emergente, o desenvolvimento do algoritmo proposto foi pouco convencional. Em primeiro lugar, foi necessário o desenvolvimento de um ambiente de testes onde o enxame pudesse ser observado em tempo real durante a sua operação. Em segundo lugar, a criação do algoritmo se deu de maneira iterativa até que fosse encontrado um conjunto de regras satisfatório.

Uma primeira solução foi encontrada modelando os agentes como máquinas de estados finitos. Este modelo de agente foi implementado com dinâmicas de comunicação indireta através de estigmergia e comunicação direta em casos de necessidade. Este método, apesar de ter apresentado bons resultados em termos de qualidade da remoção do salto de fase, necessitava ainda de um critério de parada independente do usuário. Na criação deste critério de parada, novas regras deram espaço para a criação de um algoritmo completamente diferente.

Esta segunda solução modela o agente a partir de cinco regras simples que permitem, entre outras coisas, a criação e desativação de novos agentes. Uma vez que todos os agentes são desativados, o programa chega ao fim e retorna a imagem com o salto de fase removido. A primeira destas regras afirma que se há um ou mais pixels que podem ter seu salto removido na vizinhança do agente, um deles será escolhido aleatoriamente para a operação. O agente então se move para o pixel escolhido e ganha um ponto de energia. Se não há pixels aptos a serem trabalhados, um pixel já trabalhado na vizinhança é escolhido aleatoriamente, de acordo com a segunda regra. O agente se move para o pixel escolhido e perde um ponto de energia. A terceira regra faz com que agentes que encontram dois pixels vizinhos já trabalhados mas inconsistentes entre si, marcarem estes pixels como defeituosos e desativarem-se. As duas últimas regras fazem com que agentes com energia excedente repliquem-se e aqueles sem energia desativem-se.

O comportamento esperado é que os agentes de distribuam pela imagem de maneira eficiente, aproveitando ao máximo os ciclos de processamento. Além disso, a regra de marcação de remoções duvidosas faz com que problemas de ambiguidade na remoção do salto de fase não sejam propagados por grandes regiões da imagem. Este algoritmo foi testado em diversas condições e comparado com outros métodos estabelecidos.

Os primeiros resultados foram gerados aplicando-se o enxame em imagens sintéticas sem quaisquer erros. Assim, foi possível avaliar a influência de diferentes parâmetros escolhidos pelo usuário no comportamento do enxame e qualidade dos resultados. Foi possível observar o impacto dos parâmetros de energia na densidade do enxame que, por sua vez, é importante para a correção de ambiguidades propagadas.

Em seguida, foram testadas imagens sintéticas com erros artificiais. Os resultados foram comparados com um algoritmo baseado em qualidade e um algoritmo de norma mínima. Foi observado que o algoritmo proposto foi extremamente capaz de contornar as dificuldades das imagens de maneira, produzindo resultados confiáveis. Para certas condições, os resultados foram ainda melhores que os obtidos pelo outro algoritmo baseado em qualidade.

Foram testadas ainda imagens provenientes de aplicações metrológicas reais: projeção de franjas, interferometria *speckle* e shearografia. Os resultados obtidos pelo algoritmo baseado em Inteligência de Enxame foram bastante satisfatórios, comparáveis aos métodos mais robustos. Ainda, o algoritmo proposto apresentou melhores

resultados para imagens muito ruidosas quando comparado com o outro algoritmo baseado em qualidade testado. Estes resultados atestam do potencial do método proposto em obter resultados rápidos e confiáveis.
Por fim, este trabalho foi concluído com um breve resumo destes resultados e a validação dos objetivos originais, afirmando assim o sucesso do método proposto. Foram listadas ainda algumas sugestões para avanços futuros como os testes com imagens e parâmetros de qualidade novos, a implementação de processamento paralelo e a criação de novas abordagens baseadas em Inteligência de Enxame para a solução deste problema e outros semelhantes.

Palavras-chave: Remoção do salto de fase. Inteligência de Enxame. Metrologia óptica. Processamento de franjas. Interferometria.

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
BCO	Bee Colony Optimization
CSI	Computational Swarm Intelligence
ESPI	Electronic Speckle Pattern Interferometry
FSM	Finite-State Machine
PM	Pseudo-modulation
PSO	Particle Swarm Optimization
PZT	Piezoelectric
SI	Swarm Intelligence
UP	Unwrappable Pixel

SYMBOLS

$\overline{\Delta}$	Average phase difference
$\vec{k}$	Sensitivity vector
a	Mean intensity
d	Swarm density
E	Agent energy
E_0	Initial energy
E_R	Replication energy
f_0	Carrier fringe frequency
FT	Fourier Transform
I	Wave intensity
i, j	Pixel coordinates
Im	Imaginary component
k	Window size
N	Iteration steps
N	Number of active agents
R	Correlation coefficient
Re	Real component
t	Processing time
t	Time of a wave function
T_Q	Quality threshold
T_R	Replication threshold
u	Complex wave amplitude
U	Wave amplitude
W	Wrapping operator
x	Displacement
α	Phase displacement
Δ^x	Horizontal phase difference
Δ^y	Vertical phase difference
$\Delta\phi$	Phase difference
θ	Angle of incidence
λ	Wavelength
φ	Real phase value
ϕ	Wave phase
Φ^U	Unwrapped phase value
ψ	Wrapped phase value

TABLE OF CONTENTS

1 INTRODUCTION

The correct monitoring of underground pipelines is fundamental for the oil and gas industry. The proper functioning of a transport network depends on the corrosion and stress states of the pipelines. Stress distribution, more specifically, can change over time and reach critical levels, jeopardizing the whole network. Failures might result in leaking or even explosions. In order to ensure the safety and efficiency of the oil and gas transportation, the stress state of these pipelines must be evaluated periodically.

Residual stresses are a major component of the stress conditions found in underground pipelines. This type of stress is mainly due to manufacturing processes and can be found, therefore, in several engineering applications. Its wide importance resulted in the development of many different methods of measuring it throughout the last century [1].

From among these methods, the hole-drilling technique has been selected as suitable for the measurement of stresses in underground pipelines. Recent advancements have adapted this method employing an optical assessment of the surface [2]. Hence, phase unwrapping algorithms became essential to the success of these measurements.

Phase unwrapping is a simple mathematical operation where a wrapped phase function returns to its original state by simply adding multiples of 2π accordingly. Many optical metrology methods generate a wrapped phase map, making phase unwrapping extremely important to different areas of engineering. This mathematical operation, however, becomes much more difficult with the presence of incorrect information, such as noise. Therefore, due to its importance and difficulty, phase unwrapping has brought the attention of many scientists in the past decades, resulting in a variety of methods [3].

These methods have approached the phase unwrapping problem from very different perspectives, now offering solutions better suitable to specific applications and standing on different locations of the quality vs. speed compromise scale. The most common approaches can be divided into two main groups, namely path-following and path-independent methods. Path-independent algorithms are similar to optimization techniques. They evaluate the whole image in order to find the minimum of certain parameters. This is usually a lengthy process that, however, overcomes several problems of simpler methods.

Path-following methods can be further classified into two groups: branch-cut and quality-based methods. Both aim to find a reliable integration path, the former creating blocking lines and integrating around

them while the latter employs quality parameters to evaluate which pixels should be processed first, if at all. These methods are faster but highly dependent on a reliable pre-processing of the image.

In a so far unrelated field, Swarm Intelligence has been the subject of many studies on artificial and natural intelligence. Swarm Intelligence was inspired by the collective behavior of social insects such as ants, bees and termites [4]. These insects are able to perform complicated tasks even though presenting a very simple individual behavior. The observation of this phenomenon gave place to the development of many algorithms, mainly for optimization tasks. This approach has been deemed fast, robust and flexible, which are very desirable characteristics for a phase unwrapping algorithm.

Therefore, there was the opportunity of combining those different fields in order to produce a completely novel approach to the phase unwrapping problem. Although stress measurement is the main concern of this method, a successful phase unwrapping algorithm would be able to be applied in many different methods.

1.1 OBJECTIVES

1.1.1 General Objective

This work aims to develop a novel approach to phase unwrapping based on Swarm Intelligence. This approach must result in a working algorithm that is successful in unwrapping phase maps from real metrological applications. More specifically, this method should be suitable to be employed in residual stress measurements.

1.1.2 Specific Objectives

In order to achieve a reliable method for phase unwrapping, the proposed approach must go through some necessary steps and fulfil some requirements. Firstly, a working rule set must be designed. As it is going to become clear in a later discussion, an artificial swarm must be created. This artificial swarm must be able to unwrap a phase map successfully while holding true to the fundamental aspects of Swarm Intelligence in order to maintain its usual speed, flexibility and robustness.

Later, the algorithm must be tested in a variety of conditions. These conditions should present artificial errors to help understand the swarm behavior and real metrological challenges in order to evaluate the applicability of the proposed method.

Furthermore, the algorithm should be compared to existing methods. This comparison should be able to place the proposed method in relation to available algorithms in terms of quality of results, processing times and suitability to specific phase unwrapping challenges. This work does not intend to surpass all the existing phase unwrapping algorithms. Instead, it aims to study the plausibility of employing Swarm Intelligence to this task while creating the pillars to future developments on this approach. A successful comparison with existing methods might become evidence of this approach's potential.

1.2 STRUCTURE

The work is structured in six chapters. The second chapter, following the introduction, reviews the state of art concerning fringe processing. The nature of fringes, their sources and processing methods are discussed here. In addition, the phase unwrapping problem and its solutions are thoroughly reviewed in this chapter as well.

The third chapter concerns the growing field of swarm intelligence. Initially unrelated to the central problem of phase unwrapping, this chapter will review the fundamentals of this computational approach and discuss the main applications proposed in the literature. This will serve as a reference for the proposed method.

The fourth chapter will present the proposed algorithm, starting with the unconventional design methodology. Then, the resulting algorithm is discussed in detail while being compared with the fundamental goals of this work.

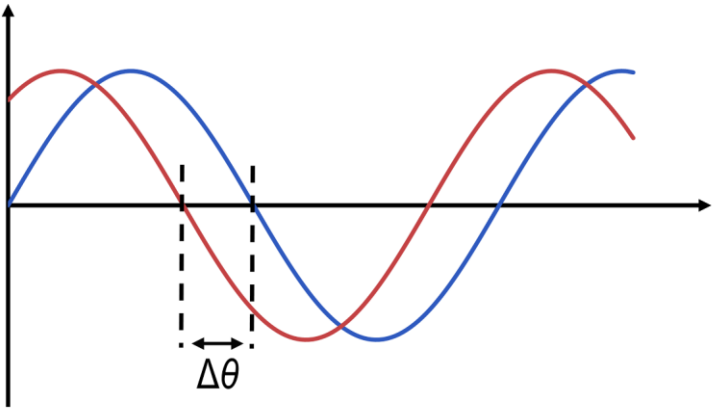
The fifth chapter will present several results for the proposed algorithm and the comparison with known methods as well, while the sixth and final chapter will summarize the main conclusions from this work and list suggestions for future developments of the proposed method.

2 FRINGE ANALYSIS

2.1 INTERFERENCE

Interferometry is one of the most powerful diagnostic tools among the optical metrology techniques [5]. At the core of every interferometric system, are the understanding of light as wave and the concept of phase. Phase can be interpreted as the initial angle of a sinusoidal function, but for interferometry, the phase difference between waves carries the most importance. Figure 1 shows the phase difference $\Delta\theta$ for two one-dimensional waves.

Figure 1 - Phase difference $\Delta\theta$ between waves.



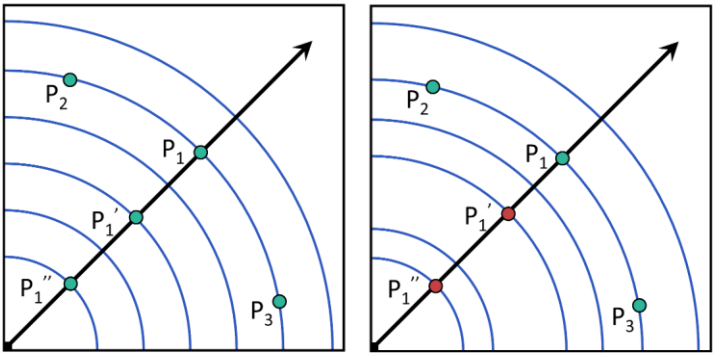
Source: Author's own work.

Phase difference alone, however, is not enough to produce interference. Coherence is another important aspect of light waves and a fundamental condition for interference. Coherence is a degree of correlation between phases of light along time or space.

Temporal coherence can be interpreted as a measure of correlation between phases of a wave along its propagation length. It can be also seen as a measure of spectral purity, i.e. for how long the wave displays a single frequency. Moreover, it can be described as the interval $[t_1, t_2]$ where the phase ϕ_2 at any time t_2 can be known given the phase ϕ_1 at time t_1 [5, 6, 7]. Figure 2a shows a temporally coherent wave. The points P_1, P_1' and P_1'' are correlated, i.e. the oscillations at P_1 are equal to those at P_1' and

P_1''. An example of a temporally incoherent wave is shown in Figure 2b, in which points P_1' and P_1'' do not correlate to P_1.

Figure 2 – Examples of a wave with both temporal and spatial coherence (a) and of a spatially coherent but temporally incoherent wave (b).



Source: Adapted from [6].

Spatial coherence, on the other hand, refers to the regularity of the wave propagation in more than one dimension, being the correlation between different points transverse to the propagation path [5, 6, 7]. It can be described as the correlation between two points of a wave front for all times. Both waves in Figure 2 are spatially coherent. In both examples, what happens at P_1 also happens at P_2 and P_3, showing a perfect correlation.

Coherence is fundamental for the observation of the interference phenomenon, as understood through the superposition principle. This principle states that whenever two electromagnetic fields overlap in space, their amplitudes are added [6, 7]. In mathematical terms, the complex amplitude of the waves can be modelled by

$$u_1 = U_1 e^{i\Phi_1} \tag{1}$$

$$u_2 = U_2 e^{i\Phi_2} \tag{2}$$

When these waves overlap, the amplitudes are added accordingly:

$$u = u_1 + u_2 \tag{3}$$

In a situation where light waves overlap, the resulting amplitude oscillates in very high frequencies, from 4.3 x 10^{14} Hz to 7.5 x 10^{14} Hz approximately, and it is not perceived by the human eye [6]. The physical quantity actually processed by humans is the light intensity that results, given perfectly coherent waves, from the following relation:

$$I = |u|^2 = I_1 + I_2 + 2\sqrt{I_1 I_2}\cos(\Delta\phi) \tag{4}$$

where

$$\Delta\phi = \phi_1 - \phi_2 \tag{5}$$

It can be inferred from Equation 4 that the observed intensity depends on the phase difference and will reach its maximum and minimum when $\cos(\Delta\phi) = 1$ and $\cos(\Delta\phi) = -1$, respectively. If the interfering waves have the same intensity, the minimum and maximum are simplified to

$$I_{\min} = 0, \tag{6}$$

for $\Delta\phi = (2n+1)\pi$ and $n = 0,\ 1,\ 2\ldots$

$$I_{\max} = 4I_0, \tag{7}$$

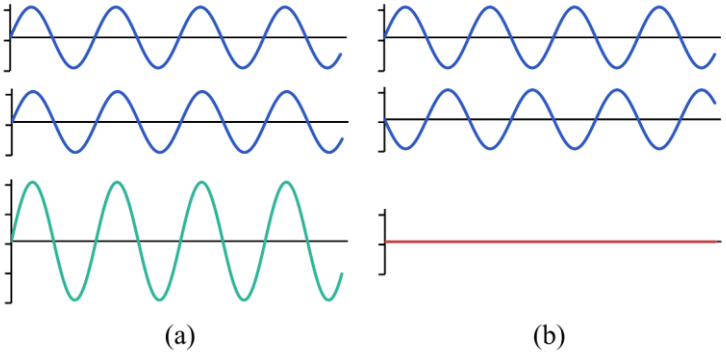
for $\Delta\phi = 2n\pi$ and $n = 0,\ 1,\ 2\ldots$

This result is illustrated in Figure 3. When the intensity of the interference is greater than the individual waves, as in Figure 3a, the interference is called constructive. The interference is called destructive if the phase difference is such that the resulting amplitude is smaller than the original waves' amplitudes, as shown in Figure 3b.

This phenomenon can be observed through Young's famous double-slit experiment, which can be performed as follows: a coherent light wave is projected in a double-slit aperture screen, where each slit behaves as a new light source. The two new waves, being related to each other, will overlap throughout space. If another screen is positioned

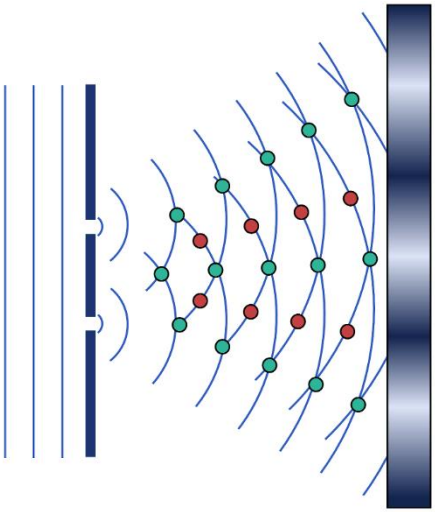
further, peaks and troughs will appear as the waves interfere. Figure 4 illustrates Young's experiment. The lines represent the wave fronts, i.e. regions with the same relative phase. As it can be seen, the result is a pattern of interference with light and dark regions similar to fringes [6, 7].

Figure 3 – Examples of constructive interference (a) and destructive interference (b).



Source: Author's own work.

Figure 4 – Young's experiment illustrating constructive and destructive interference in space.



Source: Author's own work.

2.2 FRINGES

The fringe pattern resulting from the interference of two waves depends on several physical characteristics of both the wave and the optical system. Therefore, several applications have tried to assess these physical quantities through experiments where some of these unknowns are controlled. This section will present some of the most common methods that employ fringe patterns to the measurement of surfaces and deformations.

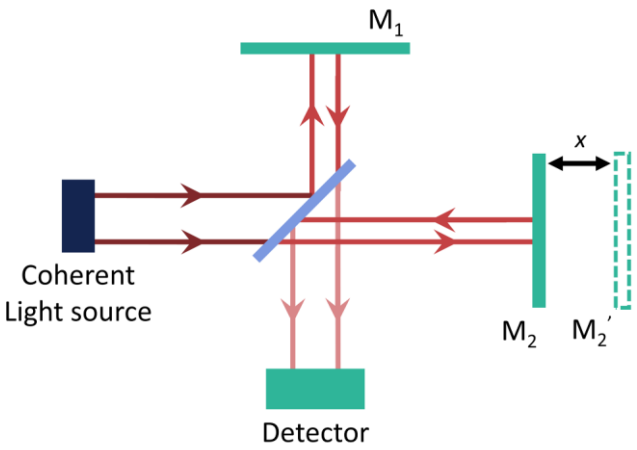
However, the formation of the fringes alone does not guarantee the complete knowledge of the underlying physical quantities. The fringes must be correctly interpreted and some methods of fringe and phase analysis will be presented further on.

2.2.1 Fringe sources

Michelson's interferometer

Michelson's experiments to inspect the existence of a luminiferous aether in the late 19th century gave birth to one of the most important interferometers in history [6]. His design of an amplitude-dividing interferometer is still employed in the measurement of smooth surfaces and lengths. Figure 5 shows the classical configuration of Michelson's interferometer.

Figure 5 – Michelson's interferometer.

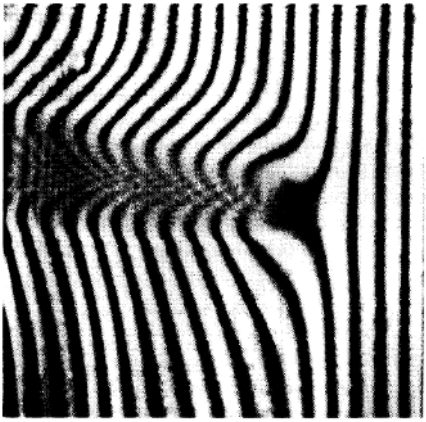


Source – Author's own work.

In this interferometer, a coherent light wave is divided into two separate beams with a beamsplitter, located at the center of the optical circuit. Both beams reflect upon mirrors and return to the beamsplitter, being recombined into a single beam producing interference. A detector is placed at the end of this circuit to register the interference field [6, 7].

One of the mirrors can be mounted on a movable platform, allowing the control over the phase difference between the interfering waves. For a displacement x, the resulting phase difference is equal to $\Delta\phi = (2\pi/\lambda)2x$. Fringe patterns arise when one of the mirrors is not perfectly flat or is slightly tilted. Figure 6 shows a fringe pattern obtained heating a small region of one mirror. Similarly, small defects in lenses and mirrors can be detected with this method.

Figure 6 – Fringe pattern from a Michelson's interferometer where one mirror was slightly heated.

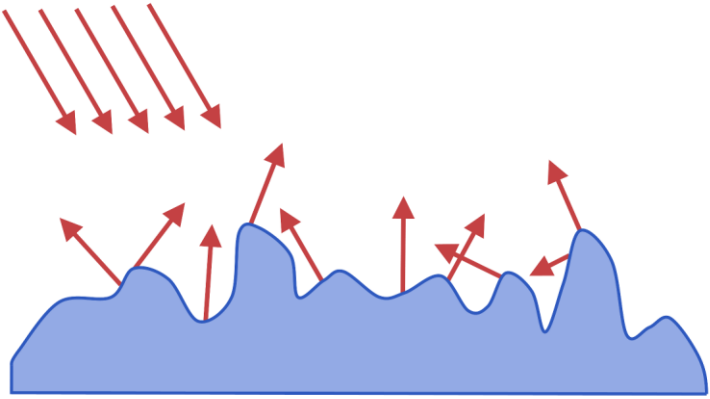


Source: [6].

Speckle interferometry

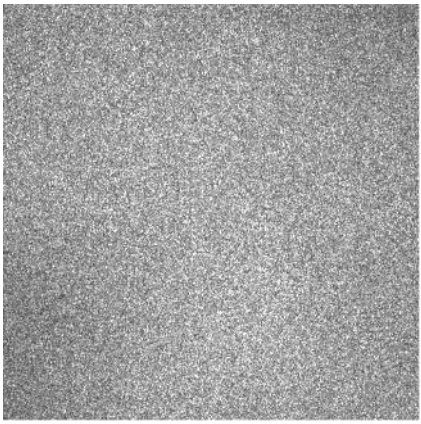
When a rough surface is illuminated by a coherent light source, a high-contrast granular pattern is observed. This structure is called speckle [5, 7]. The speckle pattern results from the interference of many waves reflected with random individual phases from an irregular surface, as shown in Figure 7. This noise-like structure, shown in Figure 8, was originally a nuisance for some optical applications, but it has been observed that positional changes on the material produce intensity changes on the speckle that can be tracked back to the deformation. This phenomenon has resulted in many measurement techniques, some of which can be found in [5, 7].

Figure 7 – A coherent light reflecting from a rough surface.



Source: Author's own work.

Figure 8 – Random speckle pattern.



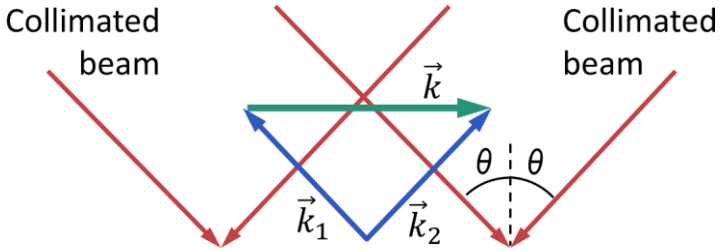
Source: Labmetro's internal production.

One important technique among those is the speckle interferometry. In this method, a reference interferometry pattern, i.e. the speckle pattern, is recorded before any displacement is applied to the measured object. When the material deforms or moves, each speckle changes accordingly, but the resulting pattern will remain random. Then, this new speckle pattern is combined with the reference and correlation fringes appear among the interference pattern.

The correlation fringes are relative to the displacement along the beam direction [5]. For the measurement of in-plane displacements, a dual

collimated beam is employed [8]. In this configuration, both beams illuminate the surface at the same angle in relation to the surface's normal. The resulting sensitivity is parallel to the material's surface, i.e. the correlation fringes will be related to the displacement along the surface. Figure 9 shows the schematic for the speckle interferometer with in-plane sensitivity.

Figure 9 – Leendertz's method for the measurement of in-plane displacement. The vector $\vec{k}$ shows the direction of sensitivity.



Source: Adapted from [8].

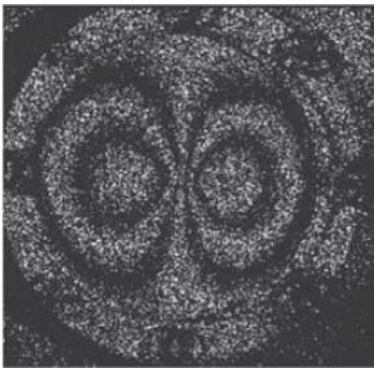
This approach can be further adapted to result in a radial sensitivity. When the laser beam is divided radially with carefully constructed diffractive optical components, it converges to a central region interfering with itself. Following the same principles that resulted in a sensitivity parallel to the surface when two beams were employed, it can be shown that this configuration will result in a radial sensitivity [9].

Shearography

In this method, the speckle pattern is brought to interfere with an identical but displaced pattern. Thus, the wave reflected from a point will interfere with that of a neighboring point. Correlation fringes appear when exposures from before and after the material displacement are combined. These correlation fringes, however, are related not to the displacement, but to the displacement differences [10, 11].

Figure 10 shows the resulting fringes from a shearography inspection. The sensitivity to the displacement differences is observed in the symmetrical patterns caused by the displacements. This technique is a powerful tool for finding surface and subsurface defects [10, 11].

Figure 10 – Shearography fringes.

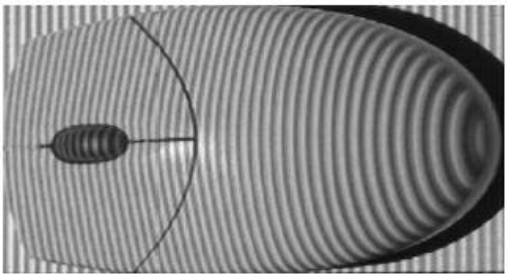


Source: [7].

Fringe projection

Fringe processing is not restricted to interferometric applications. Several optical methods use pattern projection, such as fringes, to measure three-dimensional geometries [5]. Figure 11 shows an example of fringes being projected over a surface. The measurement system consists of a projection device and an image acquisition unit. The projected fringes are structured in a known pattern, usually sinusoidal. The lit surface is registered and the image is processed by one of many fringe analyses techniques available [12].

Figure 11 – Structured fringes being projected over a mouse.



Source: [12].

2.2.2 Phase processing

For metrological purposes, the fringes must be translated into the physical quantities that caused them. This process was for a long time

limited to the intensity analysis of the fringe maps. This approach is still a viable solution for photographic records and cases where quantitative results are not needed [7].

Digital image processing, however, allowed the storage of images and the manipulation of individual pixels. Following this, phase measurement techniques started being developed. These techniques aim to extract the exact phase values from the acquired images. Both temporal and spatial methods have been studied [5, 7].

Temporal methods

Among the most common temporal methods are the phase-step techniques [13]. They are based on the displacement of the phase between the acquisitions of several images. For a set of known or constant phase displacements, the intensity maps can be converted to phase maps. This method begins at describing the intensity of a single pixel as

$$I = a + b\cos(\Delta\phi + \alpha) \tag{8}$$

where I is the intensity of the interference, a is the mean intensity, $\Delta\phi$ is the phase difference between the interfering waves and α means an experimentally controlled phase displacement. This equation contains three unknowns, a, b and ϕ, and requires at least three intensity measurements. These measurements are performed with a sequential phase displacement, thus obtaining:

$$I_1 = a + b\cos(\phi + \alpha_1) \tag{9}$$

$$I_2 = a + b\cos(\phi + \alpha_2) \tag{10}$$

$$I_3 = a + b\cos(\phi + \alpha_3) \tag{11}$$

from which derives

$$\phi = \tan^{-1}\left(\frac{(I_2 - I_3)\cos\alpha_1 - (I_1 - I_3)\cos\alpha_2 + (I_1 - I_2)\cos\alpha_3}{(I_2 - I_3)\operatorname{sen}\alpha_1 - (I_1 - I_3)\operatorname{sen}\alpha_2 + (I_1 - I_2)\operatorname{sen}\alpha_3}\right) \tag{12}$$

If the displacements α_1, α_2 and α_3 are conveniently defined, the equation can be further simplified. For example, if the phase is shifted by $\pi/2$ per exposure, giving $\alpha_1 = \pi/4$, $\alpha_2 = 3\pi/4$, $\alpha_3 = 5\pi/4$, Equation 12 is simplified to

$$\phi = \tan^{-1}\left(\frac{I_2 - I_3}{I_2 - I_1}\right) \tag{13}$$

However, this method might result in numerically unstable solutions and it is often advised to overdetermine the system with the addition of phase displacements. Further acquisitions also minimize the influence of random noise. The following techniques are some of the most common solutions:

Four-frame technique [13]
$\alpha_i = 0, \pi/2, \pi, 3\pi/2$

$$\phi = \tan^{-1}\left(\frac{I_4 - I_2}{I_1 - I_3}\right) \tag{14}$$

Five-frame technique [14]
$\alpha_i = -\pi, -\pi/2, 0, \pi/2, \pi$

$$\phi = \tan^{-1}\left(\frac{2(I_2 - I_4)}{2I_3 - I_5 - I_1}\right) \tag{15}$$

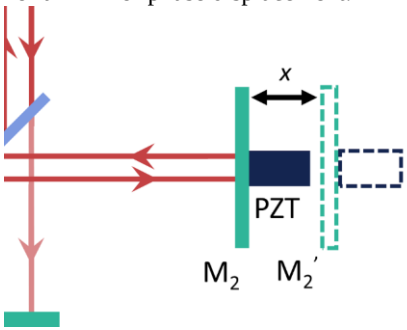
Carré technique [15]
In Carré's method, the phase shift does not need to be known, but needs to remain constant between steps.

$$\phi = \tan^{-1}\left(\frac{\sqrt{\left[(I_1 - I_4) + (I_2 - I_3)\right]\left[3(I_2 - I_3) - (I_1 - I_4)\right]}}{(I_2 + I_3) - (I_1 + I_4)}\right) \tag{16}$$

The phase shifting can be achieved experimentally by several adjustments in the optical system. Interferometers commonly displace

one mirror with a precision device such as a piezoelectric (PZT) motor as illustrated in Figure 12. For fringe projection applications, the projected image is simply replaced by a different picture with a built-in phase displacement.

Figure 12 – Application of a PZT for phase displacement.



Source: Author's own work.

Spatial methods

Among the spatial techniques, the carrier fringe method [16] is the most popular. This method superimposes a carrier fringe pattern on the interferogram or projected fringes resulting in an intensity field that can be described by

$$I(x,y) = a(x,y) + b(x,y)\cos[2\pi f_0 x + \phi(x,y)] \tag{17}$$

This equation can be written in its complex form using the relation

$$c(x,y) = \frac{b(x,y)\,e^{i\phi(x,y)}}{2} \tag{18}$$

to obtain

$$I(x,y) = a(x,y) + c(x,y)\,e^{i2\pi f_0 x} + c^*(x,y)\,e^{-2\pi f_0 x} \tag{19}$$

This spatial phase modulation carries the fundamental frequency f_0. In order to extract the phase information from the equation, the function can be transferred to the frequency domain using a Fourier transform, resulting in

$$FT[I(x,\mathrm{y})] = A(f,y) + C(f - f_0,\mathrm{y}) + \mathrm{C}^*(f + f_0, y) \qquad (20)$$

The *C* portion can be band-pass filtered and transferred back to the spatial domain through an inverse Fourier transform. Finally, the phase can be extracted from *c* applying the following equation:

$$\phi(x,y) = \tan^{-1}\left(\frac{\mathrm{Im}[c(\mathrm{x},\mathrm{y})]}{\mathrm{Re}[c(\mathrm{x},\mathrm{y})]}\right) \qquad (21)$$

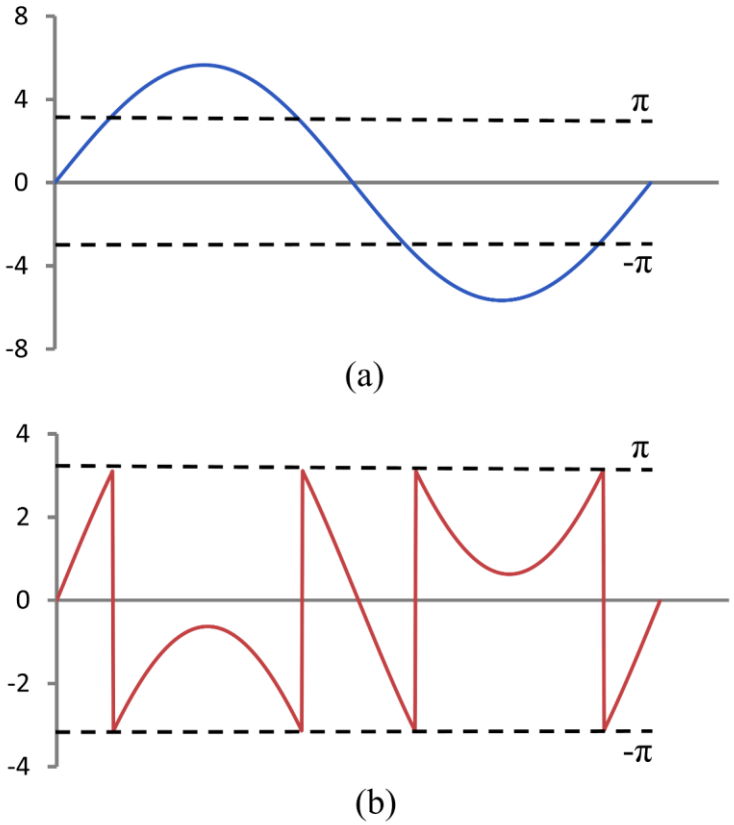
2.3 PHASE UNWRAPPING

As it can be seen from the previous section, most common phase measurement methods employ a four-quadrant arctangent function to assess the real phase values. This function's range lies in the interval [-π, π] and can be considered as a wrapping operator *W*, as in

$$\psi(t) = \mathrm{W}[\varphi(\mathrm{t})], \qquad (22)$$

where $\varphi(\mathrm{t})$ is the real phase function and $\psi(t)$ the wrapped phase obtained through the discussed phase measurement methods [3]. A simple one-dimensional example is shown in Figure 13. In this example, the original phase, which represents a physical quantity, e.g. the surface of an object, is a continuous function. The wrapped phase, on the other hand, presents several discontinuities keeping the function wrapped in the [-π, π] interval. These discontinuities appear as multiples of 2π added to the original phase.

Figure 13 – Original (a) and wrapped (b) functions.



Source: Author's own work.

2.3.1 Basic unwrap

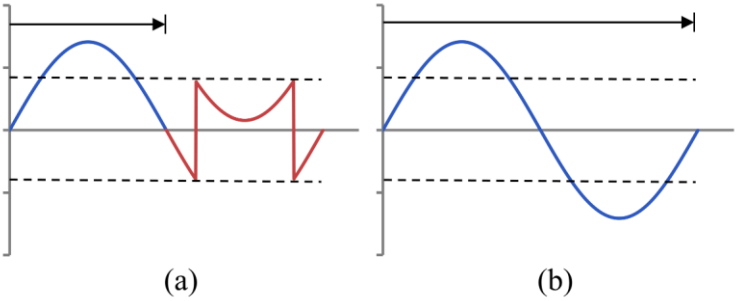
Therefore, the phase unwrapping operation consists of recovering the original phase by adding multiples of 2π to the wrapped phase [3, 17]. Figure 14 shows the same wrapped phase function being unwrapped by sequentially identifying the 2π jumps and adding multiples of 2π accordingly.

This simple method can be applied to discrete signals following the equation

$$\Phi_N^U = \Phi_N - 2\pi Round\left(\frac{\Phi_N - \Phi_{N-1}^U}{2\pi}\right), \tag{23}$$

where Φ and Φ^U are, respectively, the wrapped and unwrapped phases, and N is iteration step. For an image of a wrapped phase map, the unwrapping becomes a simple operation of sequentially comparing one pixel to the one before it until the whole image is processed. For each multiple of 2π given by the *Round* function, the wrapped phase is adjusted accordingly.

Figure 14 – Phase unwrapping being performed on a continuous function (a) and its result (b).



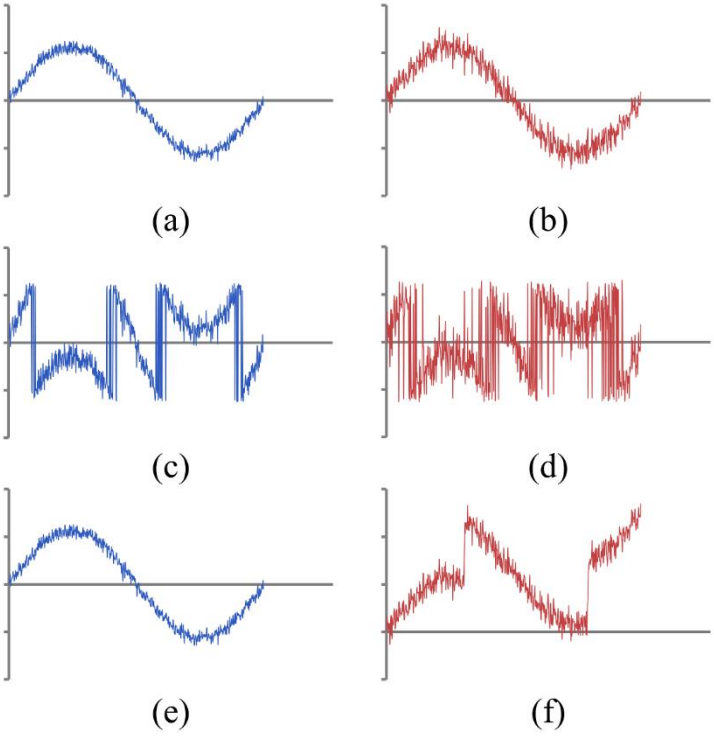
Source: Author's own work.

However, since this equation is sequentially applied pixel by pixel, the operation becomes susceptible to errors that might propagate throughout the whole image. In real images from metrological applications, these errors can be caused by noise, undersampling and ambiguity issues.

The effect of noise can be simulated and is shown in Figure 15. For this example, Gaussian noise was added to the sinusoidal function shown previously. However, the example on the right added noise with a higher standard deviation than on the left. Then, these functions where wrapped with a four-quadrant arctangent function. Finally, Equation 23 was applied in both cases to retrieve the original data.

As observable, some noise do not interfere with the unwrapping operation because they do not cause the *Round* function to return a mistaken 2π multiple. However, as the noise increases, it becomes unclear if the jumps are due to the real data or to the noise. The result on the right side is an unwrapped function that does not represent the original data after the first wrong unwrapping. If this simple method is applied to noisy images, whole portions of the phase map can become meaningless.

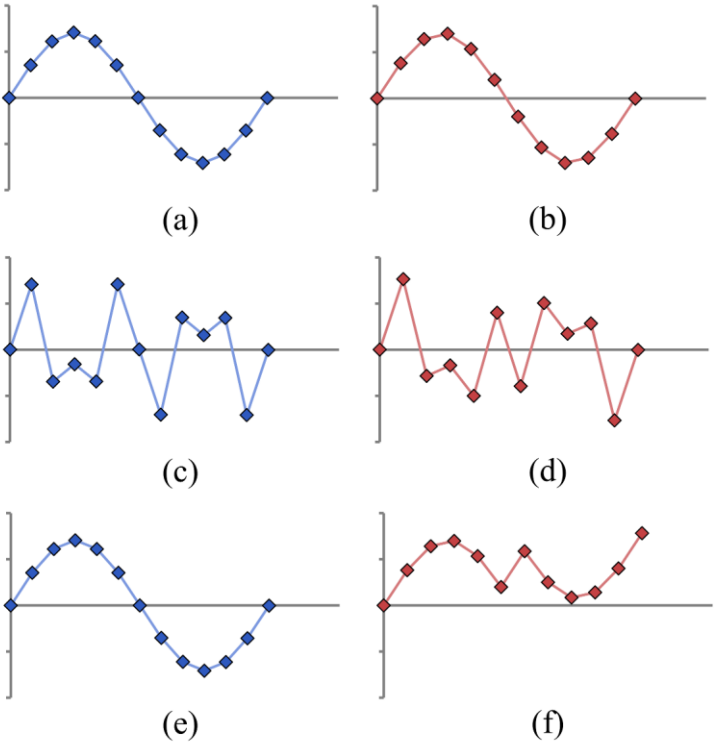
Figure 15 – Unwrapping comparison between functions with Gaussian noise with low (a) and high (b) standard deviations; (c) and (d) show the wrapped phase; (e) and (f) show the unwrapped phase.



Source: Author's own work.

Another difficulty that arises with discrete signals is the undersampling problem. Figure 16 simulates this effect with two functions sampled with 12 and 11 samples on the left and right side respectively. Similarly to the noise example, these functions were wrapped with a four-quadrant arctangent function and unwrapped with Equation 23. As shown, if the function is not sampled properly, the wrapped map will present discontinuities greater than π and the unwrapping will result in a corrupted function.

Figure 16 – Unwrapping comparison between functions sampled with 12 (a) and 11 (b) samples; (c) and (d) show the wrapped phase; (e) and (f) show the unwrapped phase.



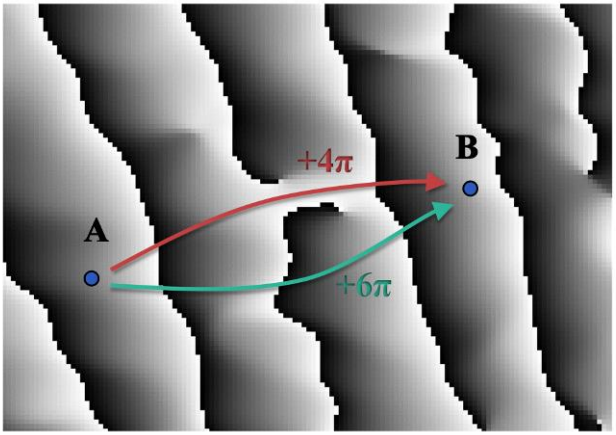
Source: Author's own work.

The ambiguity problem arises with two-dimensional arrays, such as the wrapped phase images from the aforementioned measurement techniques. As stated before, a whole image can be unwrapped by Equation 23 by applying it to every pixel sequentially. The sequence of pixels, also called the path, can be arbitrarily chosen and some phase maps show path-dependency.

Figure 17 shows an example of a path-dependent phase map. The gray levels represent the wrapped phase ranging from black, $-\pi$, to white, π. In this example, if Equation 23 is applied through the upper path from A to B, the pixel at B will have its phase added to 4π. If the lower path is chosen, however, the same pixel will add 6π to its phase. This example illustrates how the same phase map might have more than one solution. These ambiguities might result from noise-like defects in the image or discontinuities on the measured surface.

Therefore, robust phase unwrapping methods aim to minimize the negative effects of discrete sampling of two-dimensional signals whenever possible. In the past decades, several methods have been proposed and developed. The most studied and applied methods are discussed below.

Figure 17 – An example of ambiguity in the phase unwrapping of images.



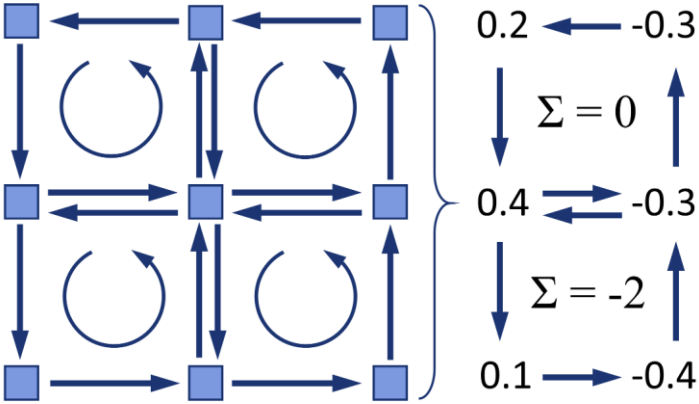
Source: Author's own work.

2.3.2 Branch-cut methods

One of the most intuitive approaches to phase unwrapping is to evaluate the path-dependency of an image. One condition for path-independency is that the integral over every closed path must return zero [3]. The simplest method for checking path-dependency in images is to perform this integral over the shortest paths possible, i.e. over a closed loop of four pixels. Figure 18 shows an example for this evaluation, with the number representing multiples of π. As shown, the integral over four pixels might return zero, showing path-independency. However, the integral might result in positive or negative sums as well. In such cases, this loop is classified as a residue with the respective sign [3].

Residue identification is the basis for the branch-cut methods. These methods aim to create branches that connect residues with opposite signs. These branches work as blockades for the path taken by Equation 23.

Figure 18 – Residue identification. The numbers represent multiples of π.

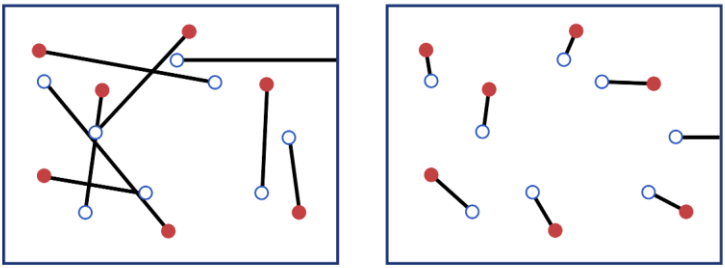


Source: Author's own work.

These branches can be connected in several different ways. Figure 19 shows an example of the same disposition of residues being connected with very different branches. The left case shows a poor choice of branches because the branches are excessively long and unnecessarily make regions unreachable by the integrating path. The solution on the right side shows a much more favorable disposition of branches, simplifying the integrating path and allowing the whole image to be processed.

Branch-cut algorithms, therefore, differ in ways of creating branches, aiming to achieve a fast and reliable method for connecting residues. Goldstein's algorithm [18] connects residues in clusters instead of in pairs. It is regarded as fast and able to create very short branches [3]. Huntley's method [19] connects pairs of residues with a simple nearest-neighbor algorithm. A modification of this method was presented by Cusack [20], along with stable-marriage and simulated annealing algorithms. A different approach, by use of the Hungarian minimization algorithm, was proposed by Buckland [21]. Moreover, methods also include minimum spanning tree [22, 23], dendriform branch-cuts [23], network programming [23, 24] and residue vectors [25] approaches.

Figure 19 – Branch cuts.



(a) (b)

Source: Adapted from [3].

2.3.3 Quality-based

Quality-based methods approach the reliability of a phase map from a different perspective. Instead of checking the path-dependency of the map and identifying residues, these methods use different information from the pixels. This information is then used to classify each pixel according to its quality. Finally, Equation 23 is applied to pixels that classify as good. These methods rely on the assumption that low quality pixels match the location of residues to some extent [3]. Below are some parameters that can be used to create the quality map.

Pseudocorrelation

Pseudocorrelation, also known as pseudomodulation, was designed to mimic the correlation parameter [3, 26]. It gives a good estimation of the homogeneity of the phase distribution. The pseudocorrelation coefficient is defined as

$$PM_{(x,y)} = \frac{\sqrt{S^2_{(x,y)} + C^2_{(x,y)}}}{k^2}, \tag{24}$$

where

$$S_{(x,y)} = \sum_{i=-1}^{1}\sum_{j=-1}^{1} \sin \varphi_{(x+i,y+j)} \tag{25}$$

$$C_{(x,y)} = \sum_{i=-1}^{1}\sum_{j=-1}^{1} \cos \varphi_{(x+i,y+j)} \tag{26}$$

and k represents the size of the k x k windows used to compute the pseudocorrelation value. Using sine and cosine functions, 2π jumps are ignored and the result is a coefficient that represents the smoothness of the phase distribution. Results closer to 1 represent a smooth region, while low values indicate a probable presence of noise.

Phase Derivative Variance

The disadvantage of the pseudocorrelation is that it classifies steep phase variations as low-quality regions, even if it correctly represents the features of the underlying physical quantity. The phase derivative variance [3] is based on the statistical variance of phase derivatives and can be obtained by

$$z_{m,n} = \frac{\sqrt{\sum (\Delta^x_{i,j} - \overline{\Delta^x}_{m,n})^2} + \sqrt{\sum (\Delta^y_{i,j} - \overline{\Delta^y}_{m,n})^2}}{k^2}, \qquad (27)$$

where $\overline{\Delta^x}_{m,n}$ and $\overline{\Delta^y}_{m,n}$ are the averages of the partial derivatives for the k x k window centered at the pixel (m, n). Moreover, $\Delta^x_{i,j}$ and $\Delta^y_{i,j}$ represent each partial derivative in the respective directions. Therefore, the resulting coefficient represents the homogeneity of the phase variance and it approaches zero for regions that represent constant phase slopes.

Maximum Phase Gradient

This quality parameter is based on the maximum value found among the phase differences in the k x k window [3]. It is defined as the greater of the two values

$$\max\left\{\left|\Delta^x_{i,j}\right|\right\} \qquad (28)$$

$$\max\left\{\left|\Delta^y_{i,j}\right|\right\} \qquad (29)$$

This parameter is based on the observations that noisy regions tend to produce large maximum phase differences. Therefore, low values of maximum phase gradient should indicate regions with low noise values.

However, it shares the same problem of the pseudocorrelation, i.e. steep phase regions result in low-quality coefficients. More quality maps are discussed compared in [27].

There are several methods for employing quality maps to guide the unwrapping algorithm. The simplest solution is to apply a fixed threshold over the image, classifying each pixel either as good or bad and then unwrapping the good pixels in any order, usually following a flood-fill sequence [28]. Thresholds can be also implemented with an adaptive behavior [29] or to be modified as the unwrapped phase map is processed [30]. More common methods, however, sort the pixels by their quality values and unwrap those who present higher quality first independently from thresholds [31, 32, 27].

2.3.4 Other methods

Flynn's method

Flynn [33] proposed an algorithm that combines both branch-cut and quality-based methods and is often referred to as a mask-cut algorithm. Flynn's method is, in a sense, the inverse of the quality-guided unwrapping, as it starts at the residues and follows the lower-quality regions creating masks. These masks grow until they encompass residues in a way that the total polarity becomes zero, i.e. the mask has the same amount of positive and negative residues. The algorithm, then, unwraps the reliable pixels using these masks as branches blocking the unwrapping path. In order to improve the unwrapped phase map, these masks might go through a morphological operation to be thinned [3].

Minimum norm

Differently from the methods discussed so far, minimum norm algorithms are path-independent and analyze the image as a whole instead of locally. These methods aim to finding the solution where the local derivatives match the real derivatives as closely as possible, similar to a surface-fitting problem [3]. An example of this approach is the Minimum L^p-norm as presented by [34], which can find weighted or unweighted solutions. These algorithms present good reliability but are computationally intensive due to their iterative nature.

Finally, phase unwrapping is a much studied subject and several other approaches can be mentioned. Some examples are cellular automata [35], calculated wrap regions [36], singularity compensation [37, 38], graph cuts [39], image segmentation [40] and wavelet algorithms [41].

3 SWARM INTELLIGENCE

Social insects have, for a long time, fascinated humans. These simple-structured animals seem capable of a high level of organization and cooperation with no apparent supervisor. This fascination has reached computer scientists as well and originated a completely new approach to old problems.

In this short section, the fundamental aspects of swarm intelligence (SI) will be discussed, followed by some examples of this behavior as observed in nature. Then, some computational swarm intelligence (CSI) applications will be presented, focusing on their potential rather than on the technicalities of each problem. This section aims to build a well-defined swarm intelligence paradigm to guide the novel approach to phase unwrapping presented later.

3.1 EMERGENT BEHAVIOR AND SELF-ORGANIZATION

Swarm intelligence, also referred to as collective intelligence, relies on a complex behavior arising from the interaction of non-complex agents. This apparent complexity or organization is called emergent behavior [42, 4]. As the term implies, this behavior cannot be found on an individual agent. Instead, the behavior emerges only with the interaction of multiple agents. Consequently, the behavior is not guided by a leader agent.

Another fundamental aspect of swarm intelligence is the limitation of each agent's awareness. Firstly, the agents do not have a global awareness of their operation or structure. Furthermore, they only gather information from their immediate surroundings. Finally, their communication is often indirect, commonly through a process called stigmergy [4]. Stigmergy is the indirect communication done through the environment. Instead of communicating in real time, one agent alters the environment so another agent will react to it in a later time. This phenomenon is found, for example, among ants and termites [4].

Emergent behavior, which is also referred to as a self-organization example, is defined by [4] as "a set of dynamical mechanisms whereby structures appear at the global level of a system from interactions among its lower-level components". Moreover, the authors in [4] list four pillars for the emergence of these structures:

1. Positive feedback, or amplification, consists of the behavioral rules that promote the creation of structures. For example, the recruitment of bees to a specific food source is an instance of positive feedback.
2. Negative feedback, on the other hand, counterbalances positive feedback, because it is necessary for a swarm to stabilize the collective behavior. The same example of bees gathering food can be mentioned. Behaviors such as competition between food sources, food exhaustion or crowding at the food source help stabilizing the collective effort.
3. Amplification of fluctuations entails a high-level of randomness in the behavior. This randomness contributes for the search of better solutions, acting as seeds.
4. Multiple interactions are at the core of intelligent swarms. Most structures arise from the interaction between agents that apply not only its individual results but also other agents' contribution to guide its future decisions.

3.2 SWARM INTELLIGENCE EXAMPLES

There are several examples of swarm intelligence to be found in nature. Some instances of natural emergent behavior are listed by [42]:

- Nest building behavior by termites;
- Task allocation in ant colonies;
- Bee recruitment through dances that result in optimal foraging behaviors;
- Communication among bacteria using molecules.

These examples cover very diverse problems and solutions. However, in order to understand how simple rules can result in an organized behavior, two examples will be discussed more thoroughly.

3.2.1 Flocks and schools

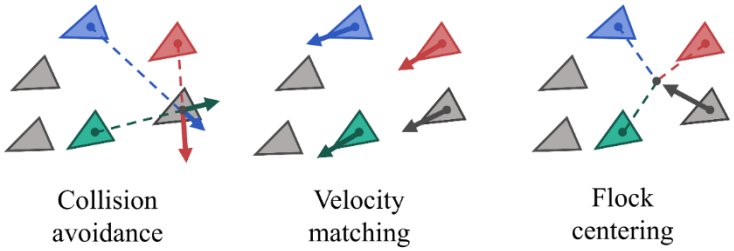
A common sight in nature is the emergent behavior found in flocks of birds and schools of fish. Even though some of these animals present higher complexity of organic structure and more direct means of communication than, for example, ants and bacteria, they present a relatively complex motion that seems organized but unguided as well.

These cloud-like structures were described and modelled by Reynolds [43], who presented the concept of boids, which is a reference to the "bird-oid" agents.

Reynolds [43] listed three fundamental behaviors that each agent should follow in order for these collective formations to arise. These rules, also illustrated in Figure 20, can be described as follows:

1. Collision avoidance: this is a static rule, meaning that only the position of the other boids will be computed. The goal is to steer away from imminent impacts with other agents.
2. Velocity matching: this is a dynamic rule because the velocity vectors of the neighbor agents are evaluated. Following this rule, each agent will try to adapt its own velocity to that of its neighbors, also contributing to avoid collisions.
3. Flock centering: also a static rule, it makes each agent try to match its position to the geometrical center of its neighbors' positions.

Figure 20 – Reynold's boids.



Source: Author's rendition of [43].

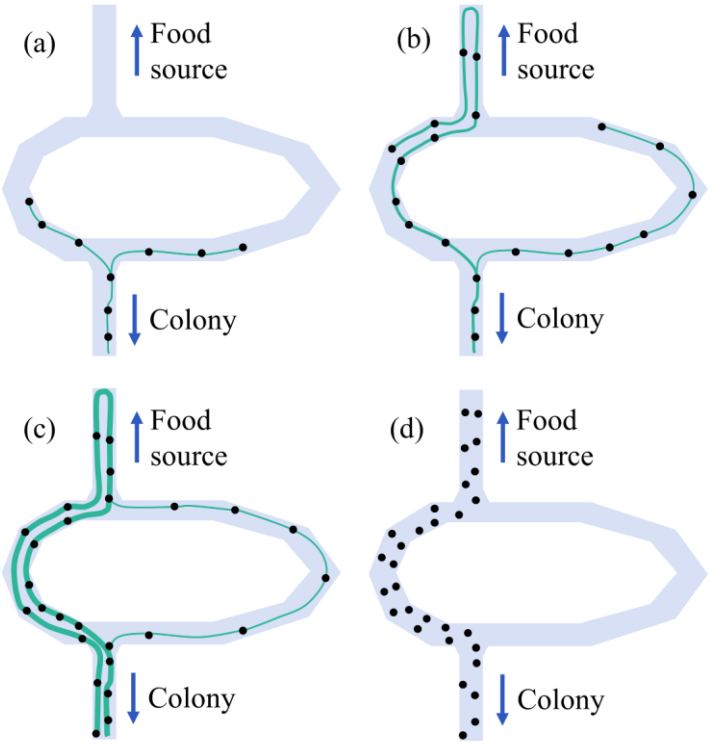
These simple rules result in an organized behavior where no agent behaves as a leader or possess global awareness of the group's condition. Moreover, these rules can be improved to include behaviors like obstacle avoidance.

3.2.2 Ant foraging behavior

The foraging behavior found among some species of ants is probably one of the most used examples for swarm intelligence. It shows great elegance and efficiency in its solution for the foraging problem. Goss et. al. [44] experimented with Argentine ants (*Iridomyrmex humilis*),

observing a natural tendency to optimize the foraging path emerging from the individual and decentralized effort of individual ants. Figure 21 shows a rendition of that work's results. The rules listed below describe how each ant behaves locally.

Figure 21 – Author's illustration of the behavior described by Goss et. al. [44].



Source: Adapted from [44].

1. Each ant searches for food while laying a trail of pheromone;
2. Upon finding a bifurcation, the ant chooses one randomly. Paths that contain a higher concentration of pheromone are preferred;
3. After reaching the food source, the ant returns with the food through its own pheromone trail.

When these rules are combined, the behavior shown in Figure 21 emerges. Figure 21a shows the first ants finding the bifurcation. The chance of following each path is 50% so the swarm divides itself equally among the options. Figure 21b shows that the ants that opted for the shortest path are able to return to the colony earlier than the other group. Returning first, these ants intensify the pheromone trail on the shortest path, making it more likable that other ants will opt for it. Finally, after a few moments, the whole swarm converges to the shortest path between the colony and the food source. As it can be seen, no information about the actual length of any path is never informed to any agent in particular, but the swarm is able to find the shortest option nonetheless. This is also a great example of stigmergy.

3.3 COMPUTATIONAL SWARM INTELLIGENCE

In order for these real life examples to be employed in computational tasks, they need to be first modelled. As proposed by [4], algorithms based on nature should mimic as closely as possible the actual properties and variables of the observed phenomenon. The next step should be the tampering the variables in order to produce efficient solutions for computational problems. For example, the ant foraging behavior as found in nature involves long times of random search and pheromone decay. After the behavior is successfully modelled, these variables should be properly adapted to computational needs.

However, swarm intelligence algorithms are not limited to representing real life examples. Based on the same philosophy, completely artificial swarms can be designed to provide novel solutions. In general, swarm intelligence algorithms are regarded as fast, robust and flexible [4]. Short summaries of swarm-inspired algorithms for optimization and image processing tasks are presented below.

3.3.1 Optimization algorithms

Most SI algorithms descend from the optimization field [45]. Some well-established algorithms are:

- Ant Colony Optimization (ACO) [42, 46] is based on the aforementioned ant foraging behavior;
- Particle Swarm Optimization (PSO) [42, 47] is a search algorithm based on the flock behavior of birds;

- Bee Colony Optimization (BCO) [48] and Artificial Bee Colony (ABC) [49] are inspired by bee's recruitment through waggle dances.

These and other swarm-inspired algorithms have been applied to a diverse set of optimization problems. Some examples are the classic traveling salesman problem [50, 48], vehicle routing [51], traffic lights scheduling [52] and supply chain network architecture [53].

3.3.2 Image processing algorithms

Nevertheless, optimization is not the only purpose of swarm-based algorithms. Some recent works have been developed on the swarm intelligence basis aiming to achieve the same level of robustness and flexibility of known models. Some works relate to image processing problems such as edge detection [45, 54, 55], image segmentation [56, 57] and object recognition [58].

While many more algorithms could be mentioned, this work does not intend to give an exhaustive overview of image processing-related works, but to show the potential of the swarm intelligence approach and the versatility of this method.

4 ALGORITHM FOR PHASE UNWRAPPING

Based on the nature of the phase unwrapping problem and the fundamental aspects of Swarm Intelligence, this section will present the design methodology used to approach a solution and two resulting algorithms. Two publications from this work's author were used in the explanation of the proposed algorithms [59, 60].

4.1 DESIGN PROCESS

Due to the unpredictable nature of swarm intelligence, i.e. the seemingly disparity between the emergent behavior and the underlying rule set, it is a difficult challenge to design a swarm starting with the desired behavior. Therefore, a different methodology was adopted by this work.

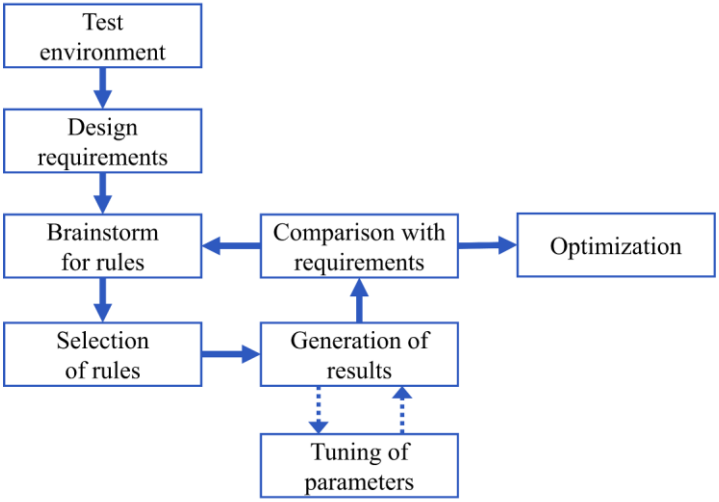
Figure 22 shows the design process for the swarm-based algorithm. Instead of progressing linearly from the expected results to the underlying principles, this method approaches the solution iteratively. However, some important steps must be taken before the actual design of the rules begins.

Firstly, an observable test environment is fundamental for the experimentation with artificial swarms. Some aspects of the emergent behavior such as spatial distribution and the creation of random paths must be evaluated in real time. This test environment does not aim to optimize the algorithm's performance, but rather to work as a canvas for the brainstorming of solutions.

Secondly, the objectives to be fulfilled by the artificial swarm must be defined with clarity. These objectives will perform the role of design requirements to guide the creative process and evaluate the success of the proposed solutions. These requirements aim to result in robustness, flexibility and efficiency in the final algorithm.

Finally, with an experimentation environment and clear goals, the actual construction of the algorithm begins. As later explained with more details, this design process is fundamentally iterative and the design requirements play an essential role of guiding the iterations. Only after the rules meet all the requirements that the algorithm can actually be optimized and go under performance tests. In the following sections, the performed work will be presented, from the creation of an experimentation environment to the final rule set.

Figure 22 – Swarm algorithm design process.



Source: Author's own work.

4.1.1 Test Environment

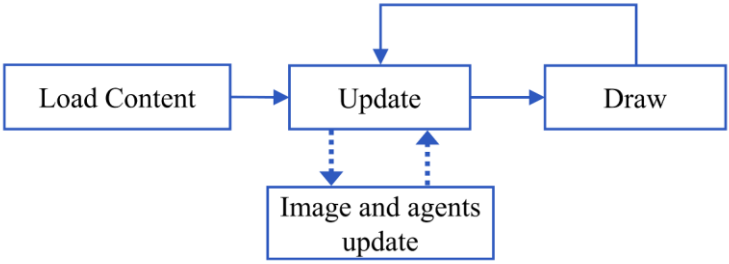
The test environment was programmed in C# using the tools from the Microsoft XNA library. XNA is a free library developed and published by Microsoft based on its .NET Framework with the purpose of promoting the development of independent games. Therefore, it contains many tools to manage and control graphic resources in interactive applications. These tools were used to build the test environment that shares the same needs of games, i.e. real time rendering and control over the program entities. Moreover, the C# language also allowed object-oriented programming enabling the creation of several instances of the same agent with ease.

Figure 23 shows the basic structure of every program built on Microsoft XNA. The program starts by loading all the necessary content in the memory. Then, the application enters an *Update* <-> *Draw* loop. The update rate can be defined by the user but is also limited by the computer's ability to process all the required information in time. During the *Update* step, all program entities are updated. In practice, this means that each instantiated object has its *Updated* method called once.

The *Draw* step consists of displaying all the graphic resources accordingly, considering the changes produced by the *Update* step. This

means that all the agents have their location adjusted in the program screen, along with any changes performed on the processed image.

Figure 23 – Update structure of the test environment.



Source: Author's own work.

4.1.2 Design Requirements

As stated above, the design requirements must cover both the reliability and efficiency aspects of the algorithm. These requirements will serve as references to evaluate the progress on the algorithm's creation. The requirements were defined as reliable unwrapping, maximum coverage, stop criteria and focused effort. These are explained thoroughly below.

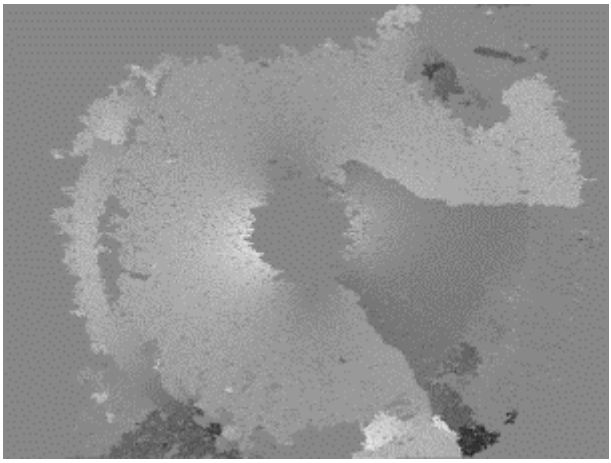
Reliable unwrapping

The most important aspect of any unwrapping algorithm is its ability to retrieve the true phase map that corresponds to the physical quantities measured by the optical system. For any application, the results of this operation must correspond as close as possible to the actual surface. However, as it was already discussed, many difficulties arise in real applications. More specifically, noise and ambiguity pose a challenge for the unwrapping algorithm.

In order to avoid letting noise-like information interfere with the results, the agents must have some method for assessing the pixel reliability. Ambiguities present a difficult challenge because the agents have limited vision. Therefore, they are not capable of inferring which side of an ambiguous unwrapped region is best for the overall operation. Since an ambiguous path might arise from reliable pixels, it is not possible to classify it as incorrect, but only inconsistent with the majority of the already unwrapped map. This evaluation must be performed, then, by a swarm composed of unaware agents. Figure 24 shows an example of

unreliable regions that were propagated from ambiguous paths. As it can be seen, the smooth, albeit noisy, surface around the center hole presents a sudden change to a darker region, which is consistent on its own but dislocated from the correct position. The program must avoid this sort of result.

Figure 24 – Example of an unwrapping operation with reliability and coverage problems.



Source: Author's own work.

Stop criteria

As already discussed, the agents in a swarm algorithm have no awareness of the global operation. Thus, a swarm does never know how much of an operation has been completed. Therefore, the method cannot employ progress-based stop criteria on an agent level.

Moreover, although a higher awareness could be used to evaluate the overall progress of the swarm, the completion of the unwrapping process would still be difficult. Firstly, there is no evaluation of the phase map prior to runtime, i.e. the program is not aware of how many pixels are reliable and can actually be processed. Furthermore, even if the number of reliable pixels were known, the ability of the program to process them all would depend on their being connect through reliable paths.

Therefore, there are many reasons for the process completion not to be used as a stop criterion. In addition, to take advantage of the swarm intelligence approach, the program should end through the interactions of

the multiple agents instead of depending on an entity with a global awareness of the operation.

Maximum coverage

As discussed on the previous design requirement, process completion cannot be used as a stop criterion. Therefore, for any other stop criterion employed, there is a chance of the process being incomplete at the end of the operation. This requirement, thus, states that a successful algorithm should unwrap most of the image. Figure 24 is also an example of an unsuccessful unwrapping due to a limited coverage of the algorithm.

Focused effort

As discussed on the pillar of self-organization, the amplification of randomness is fundamental for the swarm to find solutions successfully. Although the proposed algorithm does not involve a search for optimal solutions, the randomness will be fundamental to cope with ambiguity issues. Consequently, this randomness might results as well in un unfocused effort towards the unwrapped phase map. Therefore, it is necessary that the swarm, although random, spend most of its lifespan unwrapping pixels instead of randomly searching for them.

4.1.3 Iterative Design

The rules creation process was performed as presented by Figure 22. The first step was to brainstorm for possible rule sets. Individual rules were difficult to be tested separately because the emergent behavior only arises from the interactions of many agents and rules. Thus, different rules were designed and tested simultaneously.

Then, the algorithm was exhaustively tested. For each rule set, different configurations had to be experimented because user-defined variables presented a great impact over the emergent behavior. Only after these experimentations, the rule set could be compared with the design requirements.

This evaluation was performed only qualitatively. Firstly, because requirements such as focused effort and reliable unwrapping were difficult to measure quantitatively. In addition, these aspects were easily observed visually in the test environment. After this evaluation, the algorithm could be improved or have a new rule set designed from the start, depending on its success.

This iterative process not only helped the algorithm to be improved progressively but also gave new insight into the possible results from the

multiple interactions. Instead of creating new rules out of nothing in every iteration, past failures were used to guide future proposals.

4.2 FIRST SOLUTION

During the design process previously discussed, more than one proposal reached satisfactory results. In this section, the first successful solution is presented. This approach was based on direct communication allied with stigmergy. Although very different from the final algorithm, this solution helps understanding how the same problem can be solved by very different swarms and perhaps present an approach that can be further developed.

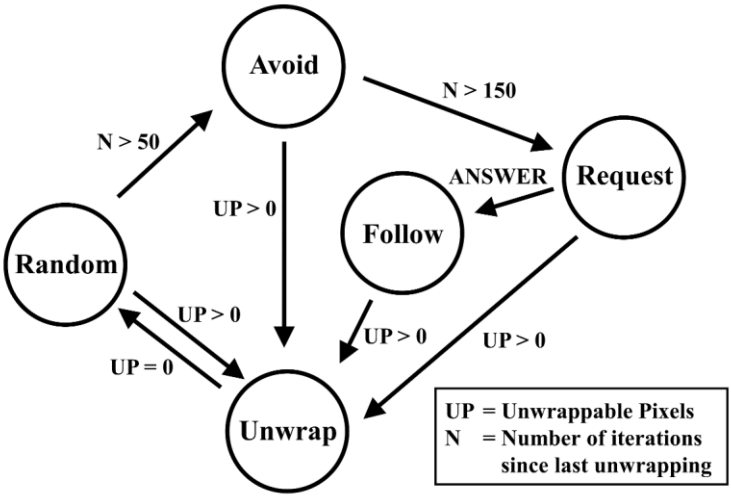
In this solution, each agent was modelled as a Finite-State Machine (FSM). A FSM is one of the simplest form of Artificial Intelligence. It can be defined by states and triggered transitions [42]. For each state, there is a set of rules that can trigger a transition to a different state. The FSM used to model the agents is shown in Figure 25. An unwrappable pixel means that it has not been unwrapped and is considered reliable according to a quality measure.

The FSM defines a simple behavior where each agent, when not able to find a pixel suitable for processing, assumes different states in order to find workable areas. This approach aims to drive the swarm to a focused effort towards unprocessed regions despite the limited awareness of each agent. The decision-making of each agent, following this FSM, can be explained as follows:

- At every iteration, the agent leaves a trace of pheromone on its current location;
- An agent stays in the first state, *Unwrap*, while there are unwrappable pixels around it;
- If the UP = 0 condition is met, the agent will search for new unwrapped pixels by walking randomly while on the *Random* state. This state intends to free the agent from locked positions and direct it towards near unprocessed regions;
- If more than 50 iterations are spent searching for unwrappable pixels, the agent enters the *Avoid* state and starts walking in the direction with less pheromone. This is a sort of stigmergy and the first rule to actually direct the agent towards a focused effort;

- If the agent is unable to find an unprocessed region, it enters the *Request* state and starts communicating with the neighbor agents. This communication consists of requesting an information about the pheromone level at the neighbors' locations. If the agent receives an answer with a preferable location, it will follow its direction.
- If at any of these search states the agent finds an unwrappable pixel, it enters the *Unwrap* state again.

Figure 25 – Finite-State Machine model of an agent.



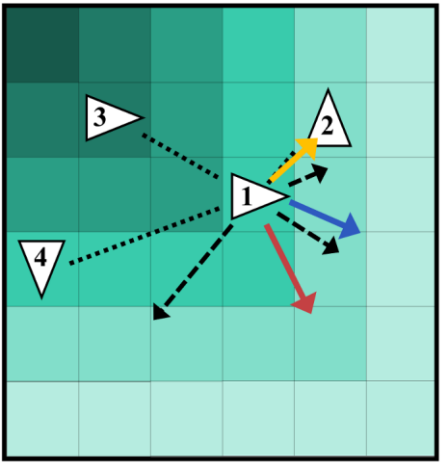
Source: Author's own work.

This model achieved good results for focused effort and was able to avoid problematic pixels following a quality evaluation that will be presented in detail with the final solution. The first difficulty arose with the proximity of the agents. Due to the communication with neighbors, the agents tended to walk very close to each other, resulting in a tightly packed swarm that was unable to search with a broad reach. The solution was to implement a collision avoidance behavior similar to that presented with Reynolds' boids [43] in Section 3.2. The result was a widely spread swarm that in practice extended the vision of a single agent to that of the whole swarm.

Figure 26 illustrates the collision avoidance and stigmergy behaviors. As shown, the urge to avoid neighbor agents is stronger for the closest neighbors. In this case, this avoidance vector will be summed to

the preferable pheromone position that comes from agent #2's location. These vectors will define the probability function that directs the random walk of the agent.

Figure 26 – Directed agents. The red arrow is the resulting avoidance vector; the yellow arrow is the preferred pheromone direction; the blue arrow is the final movement direction. The green intensity represents the pheromone level in each pixel.



Source: Author's own work.

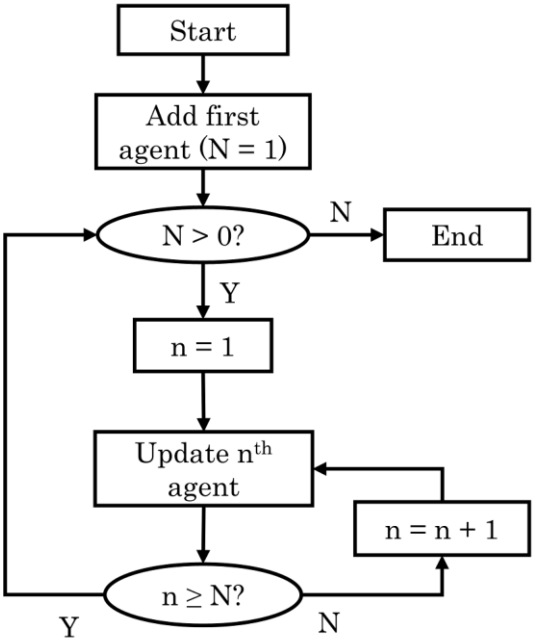
This method was capable of unwrapping difficult images while being tested in the experimentation environment. However, the program ended with a user command. This method, however, lacked a stop criterion. While designing the stop criteria, to automate and optimize this method, new rules changed the swarm behavior completely. These rules showed great potential on their own and with time replaced the present rule set altogether, giving place to the final algorithm, which is discussed on the next section.

4.3 PROPOSED ALGORITHM

Before detailing the rules and classes, it is important to understand how the agents are updated. Figure 27 shows the update structure of the proposed algorithm. The algorithm starts by placing one or more agents in a pixel chosen by the user. If the chosen pixel is not reliable for any of its neighbors, a different pixel is sought automatically. Then, the

algorithm runs through the list of active agents, calling their update function once. Each agent takes a single step obeying the rules for each update call. These updates might result in the number of active agents, N, being altered. After the current list is completely processed, an updated version is called. If eventually all agents are deactivated, the program ends.

Figure 27 – Algorithm's flowchart. The number of active agents is defined as N.



Source: Author's own work.

Pixel properties

In addition to the original wrapped phase, each pixel object must store its unwrapped phase along with three Boolean variables: *unwrapped*, *flawed* and *occupied*. The pixel quality, i.e. reliability, is not stored on the pixel object. The quality is always evaluated by each agent individually and depends on the agent's current pixel, i.e. there is a quality value for each of the eight possible directions of comparison and these values are not known prior to runtime.

The *unwrapped* Boolean simply tells if the pixel has been processed already. If true, it also means that the pixel is reliable from at least one direction. The *flawed* variable is set true if the unwrapped phase of the pixel is inconsistent with the unwrapped phase of a neighbor pixel, i.e. if the difference between two unwrapped pixels is larger than π. The *occupied* variable is defined as true at the moment an agent moves to the pixel's location, in order to avoid two agents sharing the same pixel.

Agent properties

The agents present simpler properties. The first one is a Boolean variable, *active*, that stays true for as long as the agent remains in the list that is called every iteration step by the main algorithm, as further explained. The agent is deleted from this list when this property turns false. In addition, the agents hold the property of *energy*. This attribute, stored as an integer, increases or decreases depending on the agent activity as explained in the rules, being fundamental for the success of the swarm behavior.

4.3.1 Rule set

Rule #1: Unwrap

The unwrapping of a pixel is performed every time that an agent finds a reliable pixel in its vicinity. Figure 28a shows an agent surrounded by wrapped, unwrapped and unreliable pixels. The reliability of the neighbor pixel is evaluated by a pseudo-modulation coefficient. Adapting the original pseudo-modulation presented in Section 2.3.3, to two pixels, we obtain:

$$PM_{(x,y)} = \frac{\sqrt{S^2_{(x,y)} + C^2_{(x,y)}}}{2} \tag{30}$$

where

$$S_{(x,y)} = \sin\Phi_{(x,y)} + \sin\Phi_{(x+i,y+j)} \tag{31}$$

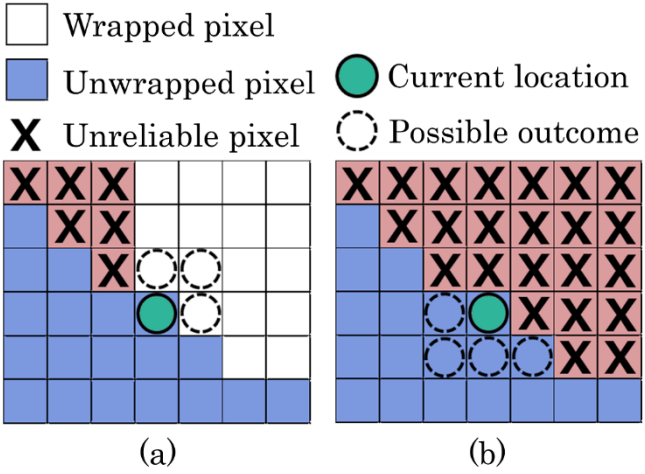
$$C_{(x,y)} = \cos\Phi_{(x,y)} + \cos\Phi_{(x+i,y+j)} \tag{32}$$

and *i* and *j* depend on the position of the neighbor pixel, varying from -1 to 1. It is important to note that the quality depends on the direction of the evaluation. Therefore, the pixels marked as unreliable in Figure 28 are simply theoretical pixels that won't yield high PM values from any direction.

In addition, Equation 30 ranges from 0 to 1 and allows the classification of the pixels based on user-defined thresholds levels. These levels, when closer to 1, restrict the analysis to the most homogeneous regions, bypassing most of the noise. However, reliable regions with a steep phase difference might be left unwrapped. Therefore, fine-tuning might be required for special cases. Although this simple comparison has yielded satisfactory results, there is still opportunity for great improvements in the pixel quality evaluation in terms of speed and reliability.

If at least one unwrappable pixel, i.e. a reliable and wrapped pixel, is found, one is chosen at random. The unwrapped phase is computed by Equation 23, replacing Φ_N for the wrapped phase of the neighboring pixel and Φ_{N-1}^U for the unwrapped phase at the agent's location. After the unwrapped phase is evaluated, the agent moves to the unwrapped pixel, gains energy, which will be important later on, and ends its current iteration.

Figure 28 – Random unwrap (a) and random search (b) rules.



Source: Author's own work.

Rule #2: Search

The first rule results in a random walk. Therefore, it is possible for good pixels to be left unprocessed due to random paths encircling good regions and moving away. In order to cover as many good pixels as possible, the agents need a rule that enables them to search for these regions. Figure 28b shows the case of an agent surrounded by unwrapped and unreliable pixels. In such cases, the agent selects an unwrapped pixel at random and moves to its location, losing energy in the process. This rule assures that each agent will never move to an unreliable pixel, even when locked. The resulting search is also a random walk and is not optimized to direct the agent to unprocessed pixels. The focused effort arises with further rules.

Rule #3: Mark

The quality evaluation employed in the first rule is unable to avoid ambiguous unwrapping operations. When two or more solutions are possible, it is desirable to promote the one that is more globally consistent. However, the agents have limited vision and only local awareness. These ambiguities need, therefore, to be solved by the collective effort of unaware agents.

The proposed rule states that if two adjacent pixels have been already unwrapped but present a difference between the phases greater than π, both are marked as flawed and the agent is deactivated. However, due to its limited vision, the agent is unable to tell which side of the ambiguity represents the best solution for the problem. By marking both pixels as flawed and deactivating, the agent stops spreading the ambiguity of its own unwrapping path and transforms the ambiguity into a barrier for movement. Figure 29 shows an example of ambiguity marking.

Rule #4: Replicate

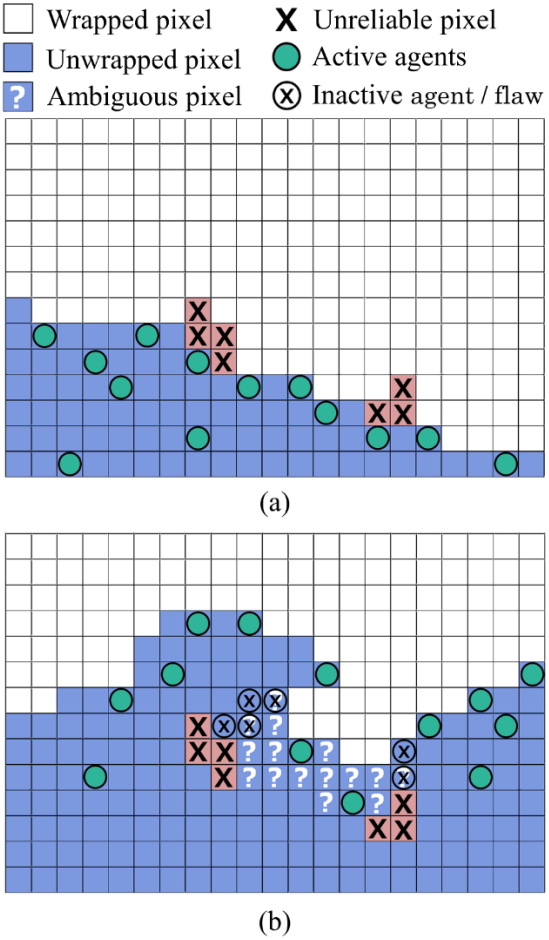
As aforementioned, a single agent is created at the start. In order to build a consistent unwrapped phase map, new agents must be created in already unwrapped regions, so the reference is the same for all operations. This rule states that if the agent has energy above a certain threshold, some of this energy is spent in order to spawn a new agent at the same location. The initial energy, as well as the energy threshold and cost for replication, are defined by the user and can be tuned for different results.

Rule #5: Deactivate

Finally, the energy attribute can also be used to direct the algorithm to its conclusion. After all energy is spent, the agent is deactivated. This

not only results in an eventual deactivation of all agents, i.e. a global stop criterion, but also avoids the processing of unnecessary random walks.

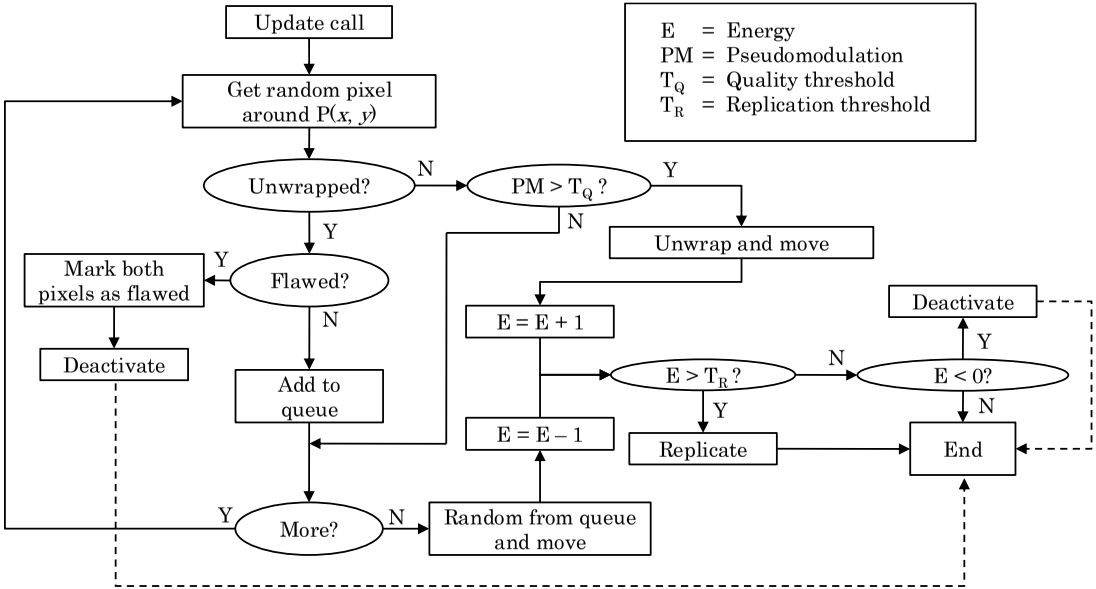
Figure 29 – Agents approach (a) an ambiguous region and mark (b) the ambiguities.



Source: Author's own work.

Figure 30 shows a flowchart representing the decision-making of each agent every time its update function is called. All the rules can be identified in the update function of each agent.

Figure 30 – Update function flowchart. Deactivation results in an instant ending of the current iteration.

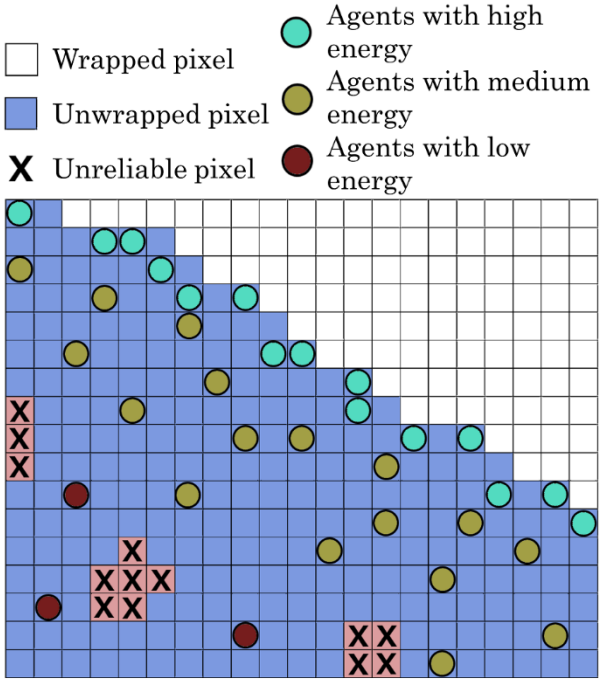


Source: Authors own work.

4.3.2 Analysis of requirements

With each rule discussed in detail, it is now possible to evaluate the original design requirements and how they relate to each rule. Firstly, the goal of covering as many reliable pixels as possible is easily met by rules #1 and #2. Although this coverage is not optimized, by walking randomly through every wrapped and unwrapped pixel, any number of agents suffices to cover all connected reliable pixels. This path-following method, however, excludes disconnected regions, as agents are unable to cross unreliable regions.

Figure 31 – Possible distribution of agents.



Source: Author's own work.

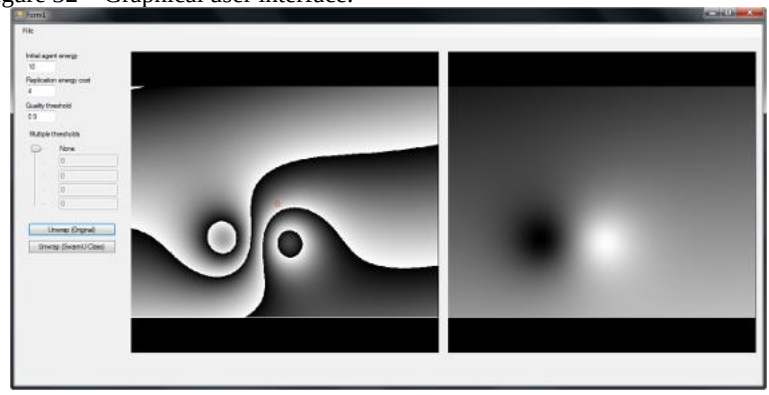
The focused effort problem is solved by rules #4 and #5 in addition to the energy gain and loss from the first two rules. Figure 31 shows a possible distribution of agents during runtime. As it can be seen, the agents closer to unprocessed regions contain more energy because they often unwrap pixels, also replicating often, creating new agents already

close to regions of interest. As some agents are left behind due to being locked in processed pixels, these spend some steps searching randomly for left behind pixels and, therefore, losing energy. Their lifespan is not long and soon they deactivate. Therefore, the replication/deactivation dynamic is able to keep most of the effort at the border, where most of the unprocessed regions is found. The additional lifespan given to the agents is able to cover small regions that were left behind, fulfilling the first objective.

4.3.3 Interface

Figure 32 shows the graphical user interface programed for the proposed algorithm. The adjustable parameters are the initial and replication energy values and the thresholds. The program allows more than one threshold to be used. In this case, when an agent has zero energy, instead of deactivating, it regains energy to lower its own individual threshold. In addition, the program also allows the selection of the starting pixel for the unwrapping operation. The left image shows the original wrapped phase map, and the unwrapped map is shown on the right side.

Figure 32 – Graphical user interface.

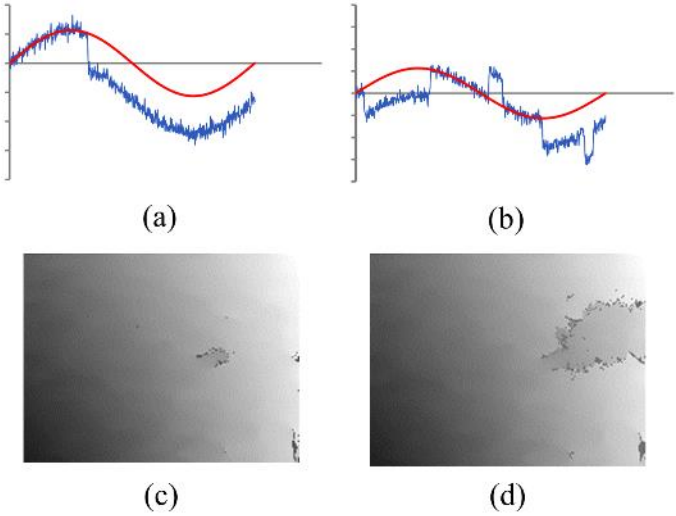


Source: Author's own work.

4.4 PERFORMANCE ANALYSIS

The quality of the phase unwrapping operation is difficult to be evaluated quantitatively. Figure 33 shows some examples of faulty unwrapping for both one-dimensional and two-dimensional cases.

Figure 33 – Comparison of unwrapping errors showing how quantitative analyses might mask the unwrapping success.



Source: Author's own work.

The first objection to quantitative analyses is that path-following methods either unwrap each data point correctly or deviate from the correct value by 2π steps, as shown in Figure 33a. In this case, the evaluation is binary: the unwrapping is either correct or incorrect. The comparison between Figure 33a and Figure 33b shows that two unwrapping operations might result in similar quantitative deviations having varying degrees of success. On the first case, there was only one unwrapping error, while on the second there were seven mistaken 2π jumps. Although a quantitative analysis is possible in this case, it does not necessarily represent the robustness of the algorithm.

A possible alternative would be to count the number of wrong 2π jumps. Although it might be a good evaluation for the one-dimensional case, it does not assess the quality of a two-dimensional unwrapping operation as well. Figure 33c and Figure 33d show the same ambiguous region unwrapped in two different cases. Although there is only one source of errors in this image, the propagation of this error was very different in each case. The case on Figure 33d resulted in many more wrong 2π jumps if evaluated two-dimensionally.

Therefore, this work will evaluate the proposed method and compare it with established algorithms only qualitatively. This same approach was employed by past works, as in [61]. The comparison will be made with the following algorithms:

- L^0-norm: This algorithm belongs to the L^p-norm methods. This iterative method is capable of correctly unwrapping difficult phase maps bypassing large ambiguities completely. However, it is a very slow method.
- Asundi's flood-fill [31]: Asundi's algorithm is a quality-based method that uses the pseudomodulation parameter to build the quality map. The unwrapping path is performed following a first-in / last-out queue.

These algorithms do not represent the latest or undisputedly best methods for phase unwrapping. However, these were the algorithms available for this purpose. The fact that these have been used extensively for different applications is evidence of their satisfactory results. Therefore, this comparison does not intend to place the proposed method among the very best solutions, but to compare it to satisfactory and well-regarded methods. The experiments will be performed on an Intel Core i7-4770 CPU at 3.40 GHz.

4.5 CASE STUDY

4.5.1 Hole-drilling measurement of residuals stress

Residual stress refers to a type of mechanical stress that is internal to a component's structure instead of being caused by external forces. Residual stresses often arise during manufacturing processes. They are characterized by being in static equilibrium and do not result in further strains in the material [62].

These stresses can be classified according to different criteria. The scale of the residual stress is one of the most common criteria. According to this scale, the stress can be classified as follows [62, 63]:

- Type I stress is also known as macroscopic stress. It extends to regions greater than the material's grain size. Therefore, this type of stress is uniform over several grains;

- Type II stress, also called microscopic, presents an order of magnitude similar to the grain size, spreading over fewer grains and usually caused by the grains' orientation;
- Type III stress affects a region smaller than the grains' size. It is often caused by small defects on the crystalline structure.

Residual stresses are usually produced by mechanical and thermal strains and are found among most of the components manufactured through techniques that include plastic deformation, thermal treatments and machining [62]. Notwithstanding, these stresses can also be found in the original material or caused by maintenance procedures. Stresses that are internal to assemblies considered as a single component can also be analyzed as residual stresses. Two riveted sheets is a simple example [64].

Awareness of residual stresses is fundamental for manufacturing process and the application of manufactured components. These stresses can be advantageous during use of the component. Compressive stress, for example, increases the component's resistance to fatigue, wear and corrosion. Tensile stresses, however, makes the component more susceptible to the propagation of cracks, lowering the component's fatigue resistance [62].

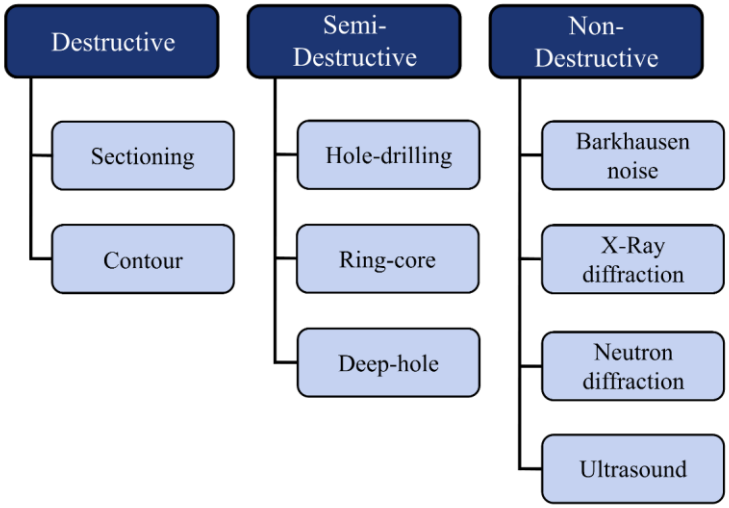
An important example is found in the oil and gas industry. Residual stresses are introduced to oil and gas pipes during their manufacturing, which includes forming and welding processes. Stresses caused by the installation and operation of these pipes are added to the already stressed state of the material. Depending on the operation conditions, these stress states may reach critical values to disastrous effects. In order to prevent this from happening, the stress condition of these pipes must be monitored.

The measurement of residual stresses poses a difficult challenge because the stresses are already applied to the component. Many techniques have been developed in order to assess the direction and magnitude of residual stresses. Figure 34 lists the most common methods. Destructive and semi-destructive techniques are based on the relief of mechanical stresses. Once the residual stresses are partially relieved from the material, the latter deforms until it reaches statics equilibrium again. This deformation is related to the direction and magnitude of the stress field, which enables the comparison between the before and after states of the material in order to assess the residual stresses. Non-destructive

methods are usually based on the analysis of the crystalline structure and are used to measure stresses closer to the component's surface.

Among these techniques, the hole-drilling technique has been one of the most employed methods, and is the most common for the measurements of stresses in oil and gas pipes. Being a semi-destructive method, it does not compromise the performance of the pipe although it removes some material from the component's surface.

Figure 34 – Residual stress measurement methods.



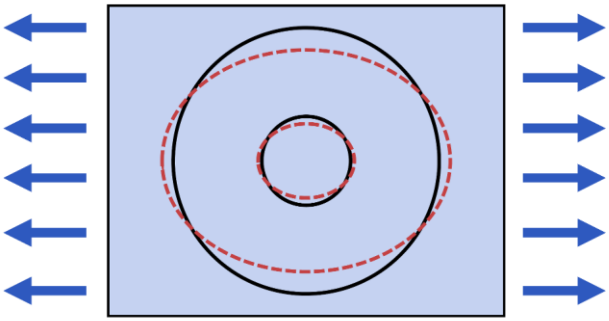
Source: Adapted from [62].

This method works based on the measurement of the surface deformation around a blind hole machined in a known location. When material is removed from a region under a stress state, the material around this removal readjusts itself until it enters static equilibrium again. This deformation is relative to the direction and magnitude of the original stress state and can be used to assess it. Figure 35 shows how the material deforms around a blind hole. The solid black line represents a theoretical circle around the region to be drilled. The red dotted line shows the expected deformation suffered by these circles after the residual stress is relieved.

This method is over a century old and it has been widely studied and normalized. It is usually regarded as simple, fast and accurate [62]. Originally, the strain measurement was performed with strain gages.

However, recent techniques have replaced the strain gages with an optical system. More specifically, the speckle interferometry discussed in Section 2.2.1, which in its digital form is called Electronic Speckle Pattern Interferometry (ESPI), has been adapted to measure radial deformation as explained below.

Figure 35 – Deformation around a blind hole under a uniform stress state.



Source: Author's own work.

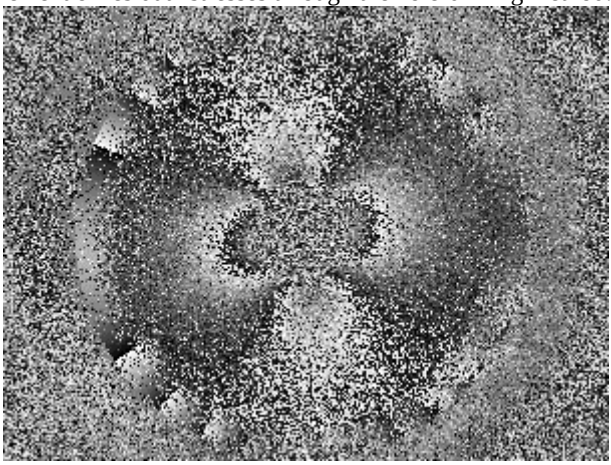
4.5.2 Speckle interferometry applied to the hole-drilling method

In Section 2.2.1, a configuration for radial sensitivity was shown. When this system is centered at the expected location of the drilled hole, the deformation will result in correlation fringes. Figure 36 shows an example of an image acquired through speckle interferometry allied to the hole-drilling technique.

This method presents several advantages over the strain gages. The speckle interferometry technique allows a real time observation of the fringes, giving an instantaneous insight of the residual stress field. In addition, this method measures the whole surface at once, instead of only three small regions around the hole. Moreover, the installation and surface preparation time is much smaller than what is required by the strain gage method [2, 1].

For phase unwrapping algorithms, this application poses a challenge due to its noisy nature and the presence of regions with no information such as the drilled surface. In this case study, the proposed swarm-based method will be compared to established methods that are currently used to interpret the fringe maps from this application.

Figure 36 – Wrapped phase map from a speckle interferometry application for the measurement of residual stresses through the hole-drilling method.



Source: Author's own work.

5 RESULTS

The discussion on the proposed algorithm showed a number of user-defined variables that must be finely tuned for the proper working of this method. Therefore, it is fundamental to understand the influence of these variables on the algorithm's performance and this section is organized accordingly.

The first experiments concern the influence of user-defined variables alone. These experiments were performed on a synthetic phase map aiming to understand how these parameters work together and to build a number of presets that could be applied to further experiments and comparisons.

After these presets were defined, the algorithm could be compared with other existing methods. This comparison was done based on phase maps obtained through real optical applications. Different applications were selected in order to present different conditions for the phase unwrapping algorithms. These comparisons will present the influence of user-defined parameters as well.

Finally, the case study proposed in Section 4.5 will be performed to evaluate the suitability of the proposed method to the stress measurement application. This section will present the results with finely tuned parameters and compare it with the current methods.

5.1 PARAMETER ANALYSIS

User-defined parameters can be classified into two distinct groups. Energy parameters are mainly concerned with the agents' life span and the replication dynamics. It is directly connected with the agent density and processing times. Threshold parameters, on the other hand, have a direct influence over the quality of results. These, however, must be tuned according to the features on the phase map and depend on the source application.

5.1.1 Energy settings

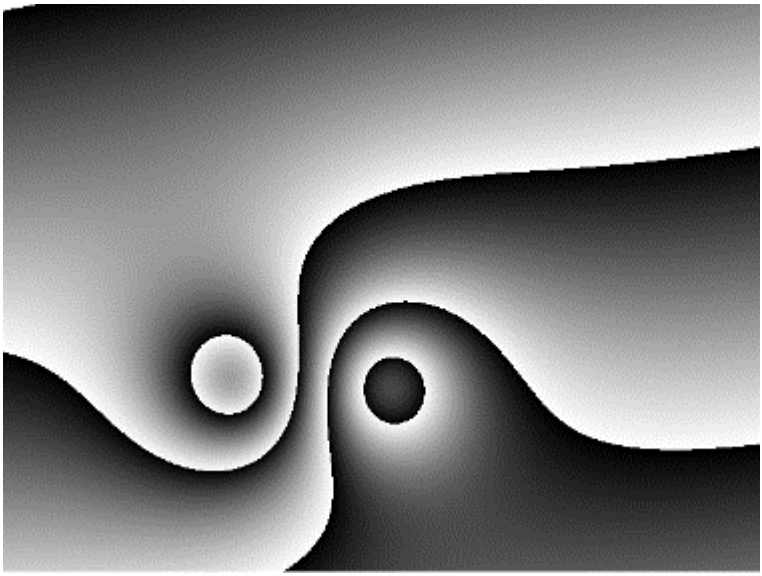
The energy settings consist of two parameters: initial energy and replication energy, represented by E_0 and E_R respectively. Initial energy is an integer that defines how much energy each agent has when created and dictates how long can an agent live without unwrapping pixels.

Replication energy is better tuned when related to the initial energy. It is defined as the amount of energy the agent needs to have

above the initial value in order to create a new agent. The replication action consumes exactly this same amount of energy, returning it to the initial value. For example, if $E_0 = 100$ and $E_R = 0.2E_0$, an agent will replicate if it reaches 120 energy, consuming 20 energy to create a new agent at the same location and with 100 energy as well. The replication threshold is connected to the agent density and it can present a high influence over the results as further shown.

The first steps towards understanding how these variables influence the swarm should be taken observing the density behavior over time. For this purpose, the synthetic image shown in Figure 37 was used for the unwrapping operation. This perfect phase map, with a 640x480 resolution, should allow the swarm to perform a complete operation regardless of the quality threshold levels.

Figure 37 – Synthetic image used for experiments with energy settings. The resolution is 640x480.

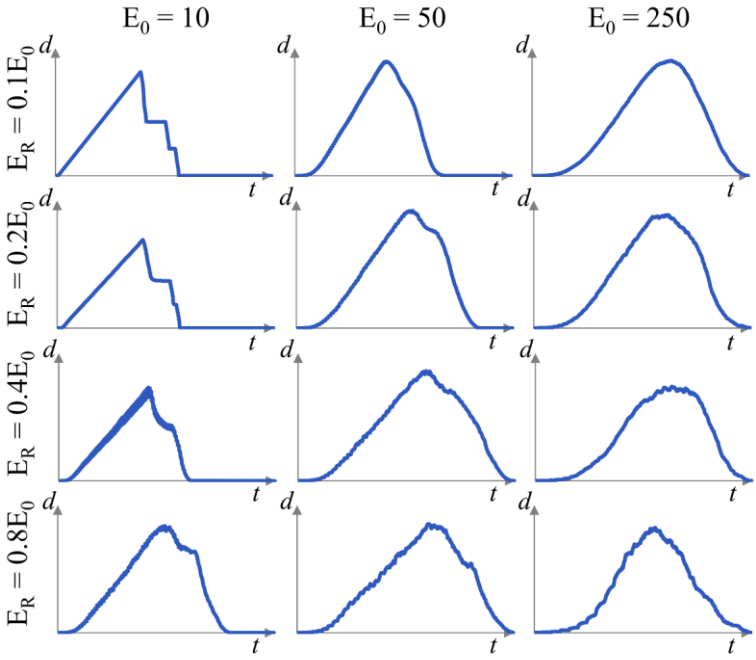


Source: Labmetro's internal production.

Figure 38 shows the results for the density over time using different energy configurations. It is important to note that the different graphs are not on the same scale. Further information can be found in the following tables. Table 1 shows the results for the peak density during runtime. Table 2 presents the total number of steps until the swarm was completely

deactivated. Finally, Table 3 shows the processing time for each configuration.

Figure 38 – Swarm density over time for different energy settings.



Source: Author's own work.

The influence of the energy settings over the swarm distribution is clearly observed in Figure 38. When agents are created with low energy and require very few unwrapping steps to replicate, the population presents a very steep slope from the start. Likewise, once the unwrappable pixels are exhausted, the swarm is almost completely deactivated in only a few iterations. In contrast, when agents have a longer lifespan, the population is better distributed over time. There is an initial inertia until the swarm grows significantly and the swarm does not deactivate so quickly.

However, the actual values of density will become more important in later results and it is, thus, important to evaluate the role of these energy settings on the agent distribution quantitatively. Table 1 shows how disparate the population can be for different settings. For low values of E_0 and E_R, the population easily surpasses 20,000 concurrent agents, while

high values produce at maximum 330 concurrent agents. These numbers are important for the swarm dynamics explained in Section 4.3.1 like the ambiguity compensation shown in Figure 29 as later experiments will show.

Table 1 – Maximum swarm population for each energy configuration.

E_R \ E_0	10	50	250
0,1 E_0	20358	13433	3612
0,2 E_0	10382	4599	1330
0,4 E_0	5854	1723	591
0,8 E_0	2513	854	330

Source: Author's own work.

As aforementioned, the graphs in Figure 38 are not all on the same scale and that means that the number of cycles required for each energy setting is also different. Table 2, when compared to Table 1, shows that higher populations achieve, unsurprisingly, the complete operation in fewer iterations. The number of cycles are relevant to the processing times, as shown in Table 3.

Table 2 – Number of algorithm iterations for each energy configuration.

E_R \ E_0	10	50	250
0,1 E_0	607	684	1469
0,2 E_0	609	819	2277
0,4 E_0	631	1147	3241
0,8 E_0	715	1719	5057

Source: Author's own work.

It is possible to observe in Table 3 that the processing times decrease dramatically for higher values of E_0 and E_R. Apparently, the higher number of iterations is overcompensated by the much lower population. It is more useful, then, to count the total number of single agent's iterations, i.e. how many times the update function is called. This is easily assessed by integrating the population over algorithm iterations.

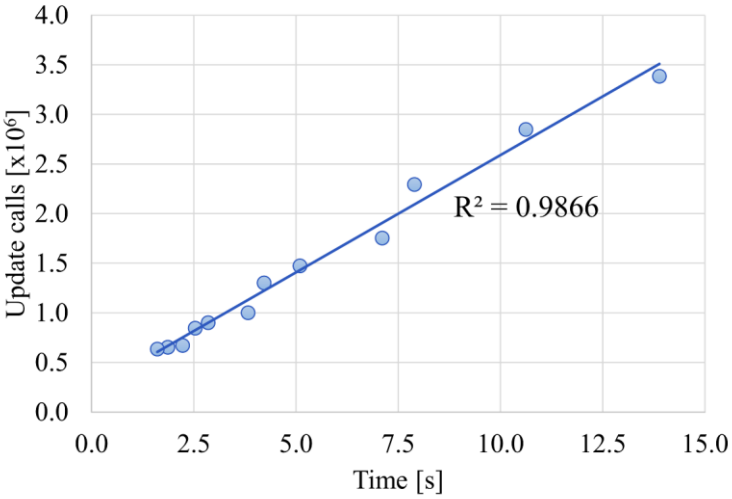
Table 3 – Total time in milliseconds for each energy configuration.

E_R \ E_0	10	50	250
0,1 E_0	13884	10618	7895
0,2 E_0	7107	5095	4214
0,4 E_0	3822	2848	2530
0,8 E_0	2226	1862	1605

Source: Author's own work.

Figure 39 shows how the number of update calls and the processing times are related to each other. It becomes clear that the integral of population over number of iterations is a good indication of how long the algorithm takes to process the whole image. Therefore, energy settings must be carefully selected in order to achieve satisfactory results in a reasonable time.

Figure 39 – Correlation between the total number of update calls and processing times.

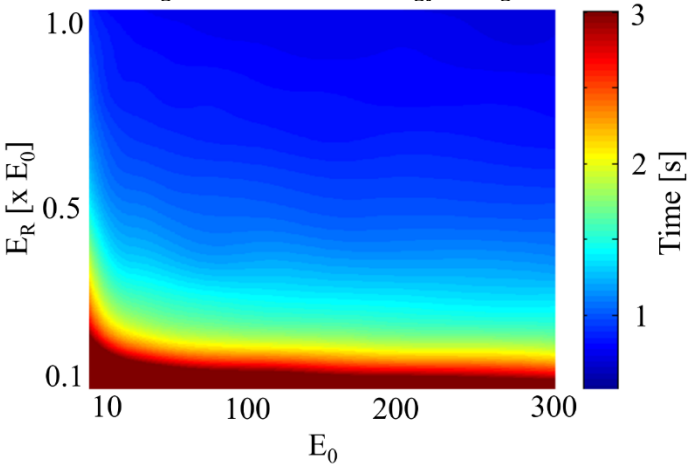


Source: Author's own work.

In order to help the selection of better energy settings, it is useful to eliminate unreasonable configurations defining some thresholds. One hundred experiments were performed, ranging E_0 from 10 to 300 and E_R

from $0.1E_0$ to $1.0E_0$. These results were interpolated and are presented in Figure 40. It is shown that low replication costs result in high processing times. At the same time, replication costs above 0.3, approximately, have little influence over the processing time. It is possible, then, to select a good energy configuration based on a time threshold. However, energy settings showed great influence over another important variable, namely the maximum coverage.

Figure 40 – Processing times for different energy settings.



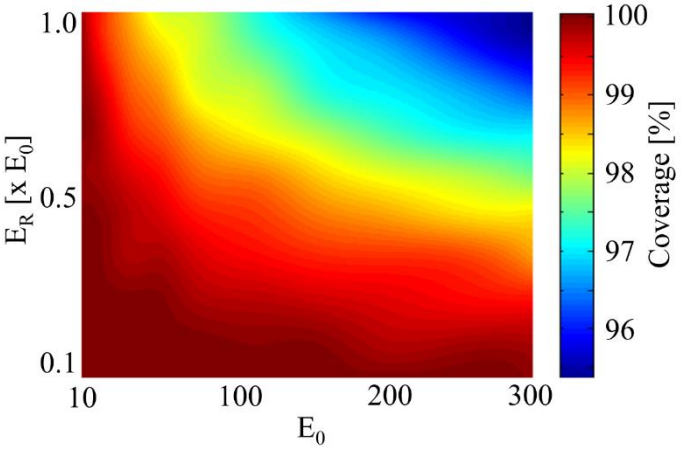
Source: Author's own work.

As explained in the design requirements discussed in Section 4.1.2, it is possible for the swarm to deactivate before every unwrappable pixel is processed. The synthetic image shown in Figure 37 presents pixels with almost perfect pseudomodulation. Therefore, unprocessed pixels are due only to a faulty coverage by the swarm. It was observed that energy settings had a great impact on the swarm's ability to cover the whole image and the same experiments shown in Figure 40 were used to assess this impact.

Figure 41 shows the influence of the energy settings over the maximum coverage, indicated as a percentage of the image pixels that were actually processed. It can be observed that higher values of E_0 and E_R produce worse coverage. When compared to the previous discussion, it is clear that smaller populations are unable to cover the whole image as successfully as higher populations, even though the former require more iterations. It is probably due to the overall density, since denser swarms

are less likely to leave unprocessed pixels behind after their random walks. Although some applications are not influenced by the loss of a small percentage of data, a better coverage is desired if the processing times are reasonable, even if only for aesthetic purposes.

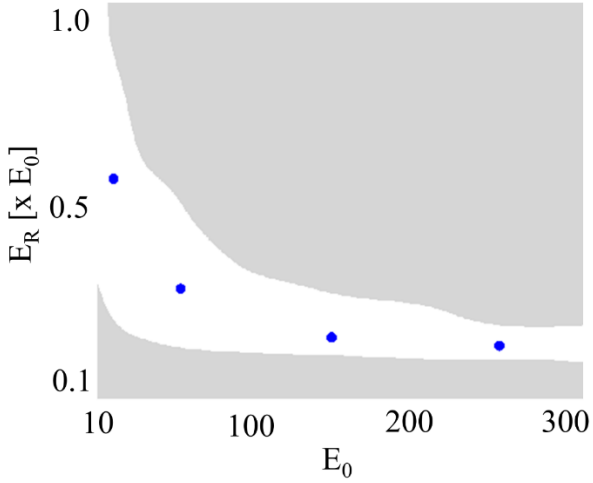
Figure 41 – Maximum coverage for different energy settings.



Source: Author's own work.

Therefore, if thresholds are set for both processing times and maximum coverage, it is possible to obtain a narrow range of possible energy configurations. Figure 42 shows the possible range, represented in white, of configurations that result in processing times faster than 2 seconds and coverages greater than 99.5%. As previously stated, the population size will influence some collective dynamics and it is useful, then, to define presets that result in a variety of population sizes while obeying the time and coverage threshold. These presets are marked in Figure 42 and detailed in Table 4.

Figure 42 – Energy settings range, in white, for processing times lower than 2 seconds and maximum coverage above 99.5%.



Source: Author's own work.

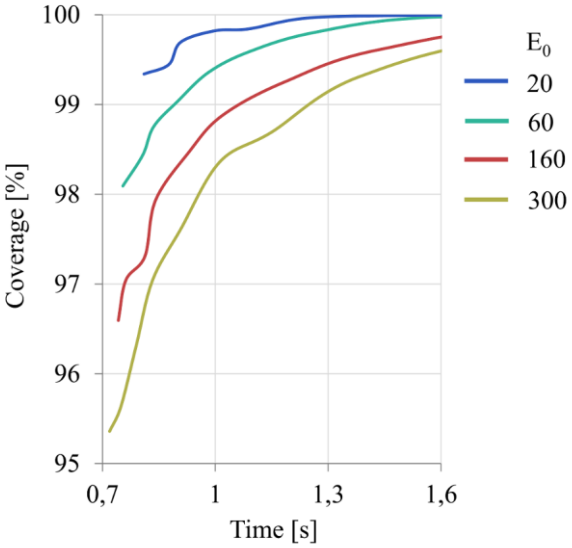
Table 4 – Energy settings presets and the corresponding results from the synthetic image.

Preset	E_0	E_R	Time [ms]	Coverage [%]
A	20	0.6	1090	99.85
B	60	0.35	1400	99.90
C	150	0.24	1680	99.85
D	250	0.22	1740	99.65

Source: Author's own work.

Finally, it is often preferable to choose larger populations. Figure 43 shows how processing times are related to maximum coverage for different energy settings. The energy settings are only represented by E_0 as the E_R values are indirectly represented by the processing times. From the results in Figure 43, it is observed that lower values of E_0, and therefore larger populations, produce better coverage for the same processing times. Similarly, larger populations require less time to achieve the same coverage. Smaller populations will be preferable only when ambiguity compensation becomes more necessary.

Figure 43 – Coverage values against processing times for different energy settings.



Source: Author's own work.

5.1.2 Randomness

As previously discussed, randomness is a fundamental aspect of swarm-based algorithms. Therefore, the proposed method often presents different results for different runs under the same configuration. Most of the time, these differences are insignificant when compared to the overall result. For smaller images with more difficult ambiguous regions, these differences might compromise the reliability and usefulness of the results. It has been also observed that these differences depend on the swarm density. Very dense populations will produce the same results more consistently because the agents have little room to walk randomly. Sparse populations, on the other hand, allow these random walks to differ more from one experiment to the other.

Therefore, the results presented on the next section will be selected from a set of three sequential experiments at most, if the first ones are not satisfactory. The variability of the results will be discussed only qualitatively.

5.2 ALGORITHM COMPARISONS

In this section, the proposed method will be tested under several different circumstances presenting different obstacles to a phase unwrapping algorithm. These tests not only intend to evaluate the overall success of the proposed method but to compare it with existing and established algorithms, as explained in Section 4.4.

Processing times will be presented for the swarm-based algorithm. However, the other algorithms were implemented in different programming environment with different levels of optimization, making it difficult, therefore, to compare their performances honestly. Still, it is important to note that minimum norm algorithms, such as the tested L^0-norm method, are slow and usually take a few minutes to unwrap average-sized images like the ones used in this work. On the other hand, quality-based algorithms are usually very fast, unwrapping these images well under a second.

Therefore, processing times will be provided in order to understand the influence of the energy parameters and the compromise between performance and quality of results. Finally, both path-following methods, i.e. the quality-based and swarm-based algorithms, will start the unwrapping operation at the center of the image unless otherwise stated.

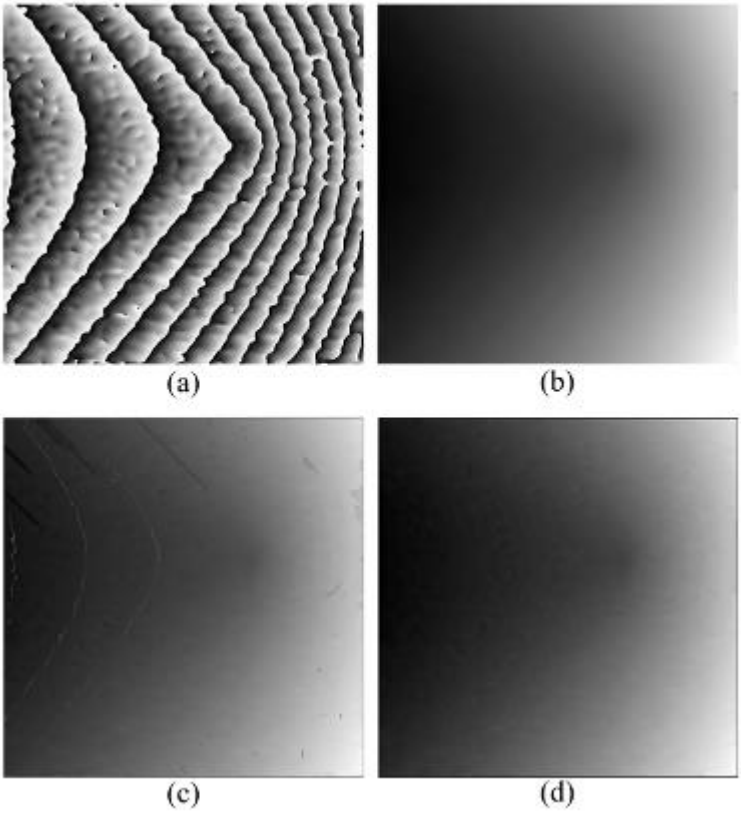
5.2.1 Synthetic images

Before experimenting with real metrology applications, it is useful to further experiment with synthetic images. This allows testing the proposed method under artificial circumstances that highlight a certain aspect of phase unwrapping. For example, ambiguity errors can be artificially produced to create a difficult unwrapping task to be solved by different methods and configurations.

Figure 44a shows an example of artificial ambiguity errors. This image was firstly created free of errors by wrapping Figure 44b that also serves as a reference for the unwrapping results. Later, errors were introduced in the image to create difficult ambiguous regions.

Figure 44c shows the phase map unwrapped by the quality-based algorithm. The propagation of several small errors can be observed. In addition, some regions with 2π jumps presented lower quality and resulted in white lines crossing the image. Overall, the quality-based method was successful, despite these errors being always present.

Figure 44 – Synthetic image (512x512 pixels) with artificial errors (a), its unwrapped reference phase map (b) and the unwrapped phase map obtained with quality-based (c) and L^0-norm (d) algorithms.



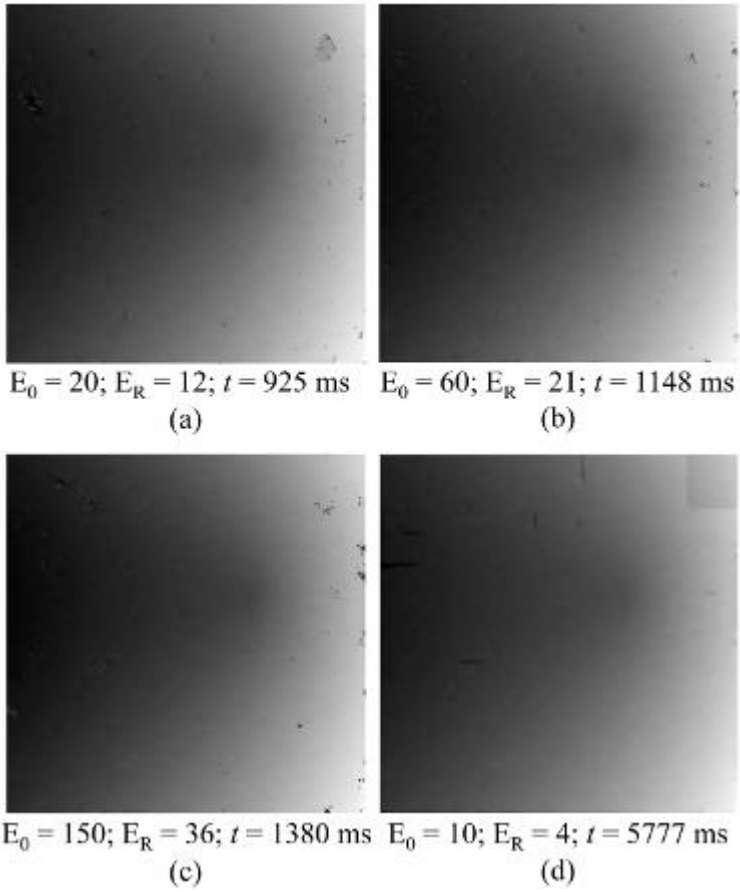
Source: Author's own work.

Figure 44d shows the solution from the L^0-norm method. This method is very successful in unwrapping this phase map because it does not follow integration paths. Therefore, ambiguity errors are almost completely ignored. This approach represents the other end of the speed vs. reliability compromise spectrum.

The results from the swarm-based algorithms are shown in Figure 45 along with the energy settings used and the processing times. Three presets defined in Section 5.1.1 were employed first. The first preset, which results in the higher population, was tested first and the best result from 3 consecutive runs is presented in Figure 45a. It can be observed that

the smaller ambiguities were easily suppressed by the collective effort, confirming the expectations on the rules interactions discussed in Section 4.3.1 and shown in Figure 29. This suppression is evident when the results are compared with the other path-following method shown in the previous figure. A larger propagation, however, can be found in the upper right region of the image, where a much more difficult ambiguity is located. Due to its high density, the swarm is unable to suppress this error propagation quickly.

Figure 45 - Phase maps unwrapped by the swarm-based algorithm with different settings. All operations used a single 0.9 threshold.



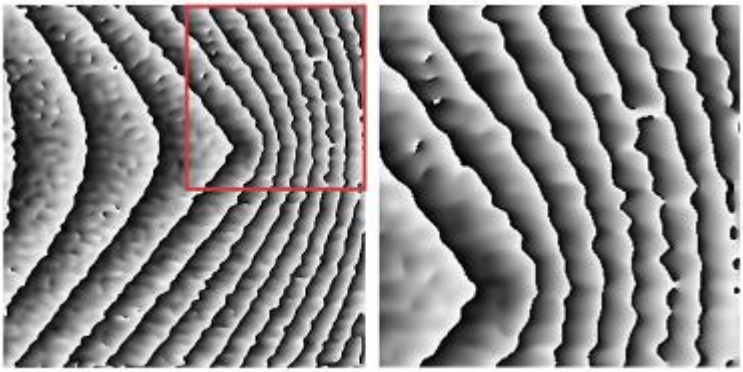
Source: Author's own work.

The influence of the swarm density is clearly shown in Figure 45b and Figure 45c. With energy settings resulting in less simultaneous agents, less agents will pass through ambiguities and these are easily corrected by the collective behavior. In addition, the processing times are not much longer for these settings.

In order to highlight the influence of swarm sizes, Figure 45d shows the results for a much larger population. The propagation of the ambiguities becomes much more intense and the variability between consecutive runs becomes smaller. Moreover, these settings result in an unnecessarily high time cost.

Population size, however, is not the only parameter to influence the propagation of ambiguities. The swarm direction changes how these ambiguities are dealt by the agents and it depends mainly on the starting position. To better show this, a smaller region of the synthetic image was selected as shown in Figure 46. This region presents the larger ambiguity in its upper right corner along with smaller errors.

Figure 46 – Selection of a difficult region from a synthetic image with errors.

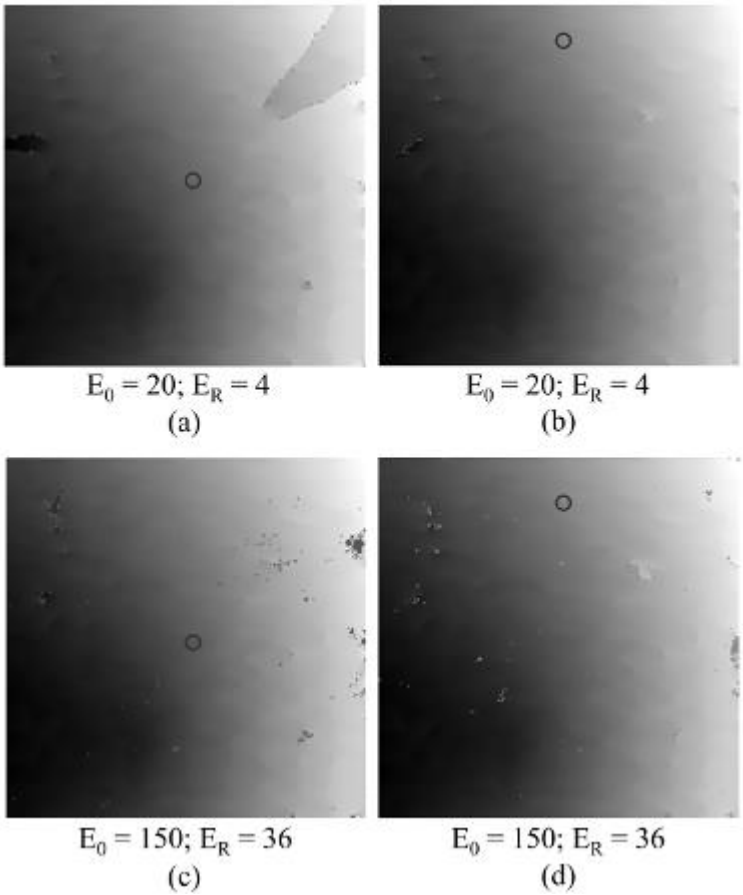


Source: Author's own work.

Figure 47a and

Figure 47b show the unwrapped phase maps for the same settings but with two different starting points. It is observable that ambiguities are easily suppressed for the second swarm positioning. If the ambiguous line is parallel to the swarm direction, fewer agents will cross it and the ones that do are easily corrected by the others. If ambiguities are obvious, changing the starting point is an easy solution to problematic phase maps.

Figure 47 – Unwrapped solutions from the swarm-based algorithm. The circles mark the starting pixels. All operations used a single 0.9 threshold.



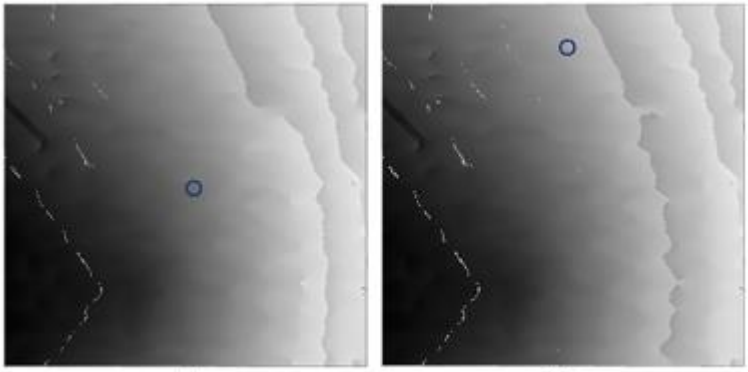
Source: Author's own work.

It can be also observed that the dependence of the starting point is different for different population sizes. Figure 47c and Figure 47d show the same comparison but with a smaller population. Since the ambiguity was easily suppressed on the first case, the starting position shows little influence. Therefore, changing the starting point is a good solution for images that require swarms with a higher density.

Figure 48 shows this same experiment performed with the quality-based algorithm. Although this algorithm also depends on the starting

point, both results are greatly corrupted. This is mainly due to the size and proximity of the larger ambiguity, since its propagation was much more intense than in Figure 44d. Notwithstanding, these results show the potential of the swarm-based approach to produce reliable results in relatively fast processing times, especially when compared to existing methods.

Figure 48 – Quality-based solution for the phase unwrapping of an ambiguous region. The circles mark the starting pixels.



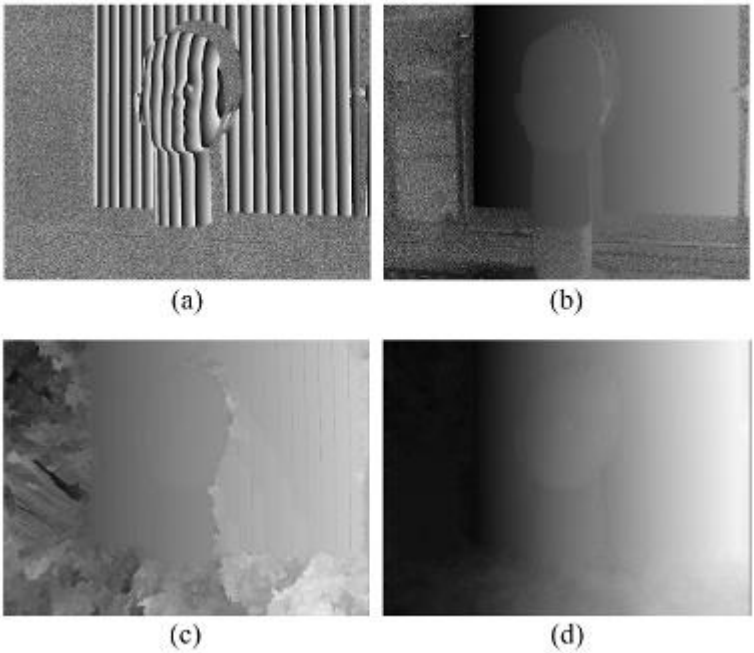
Source: Author's own work.

5.2.2 Fringe projection

Although synthetic images provide good test subjects for phase unwrapping algorithms, real applications hold many features that are difficult to recreate artificially. These features not only pose new challenges for these algorithms but they represent the actual obstacles to be overcome by any unwrapping technique intending metrological applicability.

Fringe projection, for example, is a common measurement technique that requires a phase unwrapping operation. Two objects were measured with this method and the resulting phase maps were used for the comparison of the algorithms. Figure 49a shows the wrapped phase map for the measurement of a face mannequin.

Figure 49 – Wrapped phase map (1624x1224 pixels) from a fringe projection application (a) and the phase unwrapped using graycode (b), quality-based algorithm (c) and L^0-norm algorithm (d).



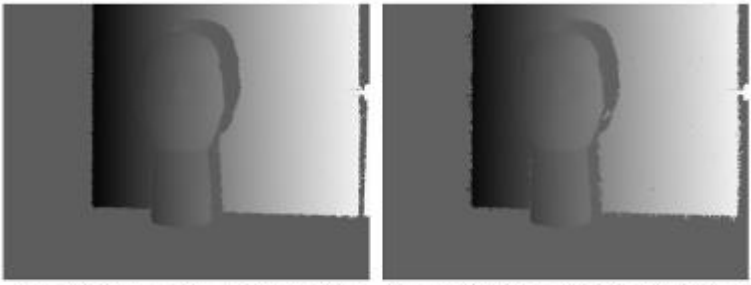
Source: Author's own work.

In fringe projection applications, it is possible to include further information in order to guide the unwrapping operation. The projection of graycodes informs the unwrapping algorithm in which fringe order each pixel is, removing all sources of ambiguities [65]. This method was employed to obtain a reference phase map, which is shown in Figure 49b.

Conventional algorithms, however, must rely only on the information provided by the phase map. Figure 49c shows the unwrapped phase map from the quality-based algorithm. It can be observed that although the image is relatively noise-free, this algorithm was unable to provide a smooth unwrapped phase map. In addition, it is possible to observe inconsistent 2π jumps at the right side of the mannequin. Figure 49d shows the smooth results from the L^0-norm method. Both these methods, however, are incapable of correctly unwrapping surface discontinuities without further information, as later shown.

The results from the swarm-based algorithm are shown in Figure 50. Unlike the synthetic images, these phase maps present whole regions without information, which are left unprocessed by the swarm. In addition, the swarm size seems to be irrelevant to the quality of results, affecting only the processing times. There is no apparent benefit of using smaller swarms as these were slower to process.

Figure 50 – Fringe projection phase map unwrapped by the swarm-based algorithm.



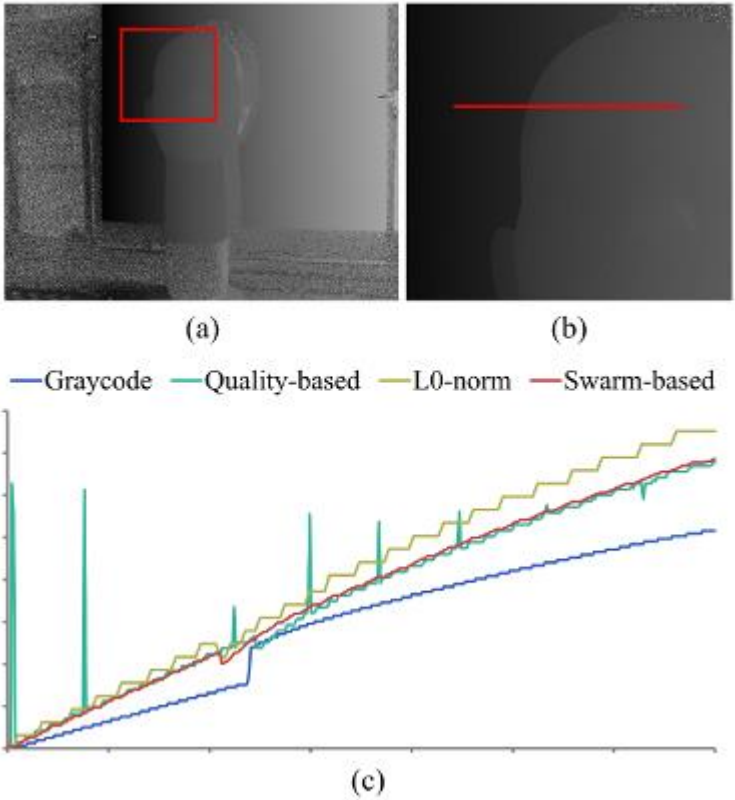
Source: Author's own work.

As already mentioned, these algorithms are unable to interpret surface discontinuities without further information. Figure 51 shows the intensity profile from an arbitrary path for all the tested methods. These profiles were normalized and the comparison is purely qualitative. It is clear that no algorithm was capable of interpreting the discontinuity between the mannequin and the background surface. The swarm-based algorithm was, however, able to provide the smoothest results.

Another experiment was performed with a smooth surface. The wrapped phase map is shown in Figure 52 along with the phase unwrapped by the same methods. Once again, the quality-based algorithm had trouble unwrapping the phase jumps while the L^0-norm provided a smooth phase map.

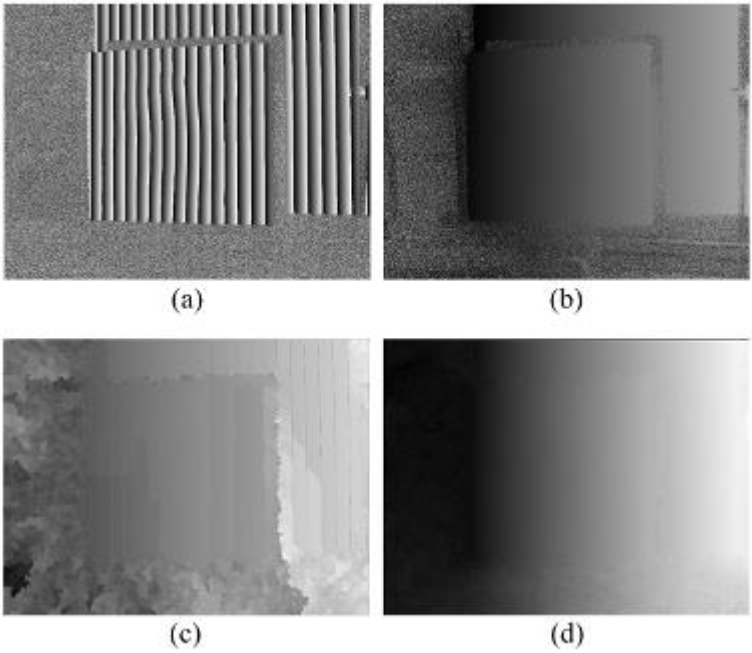
Figure 53 shows the result from the swarm-based algorithm. Similarly to the first example, the algorithm easily unwrapped the phase map. There was no need to employ smaller swarms. These results also show some potential for application of the proposed method.

Figure 51 – Comparison of the intensity profile along the red line (b) for different unwrapping techniques.



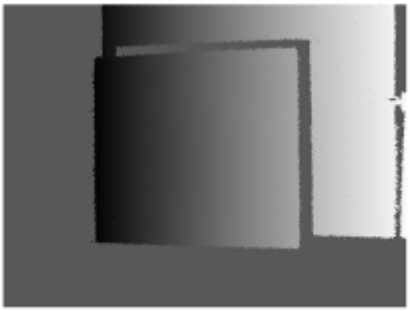
Source: Author's own work.

Figure 52 - Wrapped phase map (1624x1224 pixels) from a fringe projection application (a) and the phase unwrapped using graycode (b), quality-based algorithm (c) and L^0-norm algorithm (d).



Source: Author's own work.

Figure 53 – Fringe projection phase map unwrapped by the swarm-based algorithm.



Source: Author's own work.

5.2.3 Shearography

So far, all the cases presented relatively noise-free phase maps. Single thresholds enabled the swarm to reach almost the complete image. In addition, swarm sizes had an impact only on the ability to suppress ambiguity propagations and the processing times. Several optical applications, however, must deal with the presence of noise.

Shearography, as explained in Section 2.2.1, uses a random speckle pattern to produce correlation fringes that can be tracked to deformations on the measured surface. This noise-like pattern poses a challenge to phase unwrapping algorithms and allows further comparisons between the proposed approach and established methods. Furthermore, it also allows the experimentation with different algorithm parameters to better understand the collective behavior.

It is important to mention that several filters can be applied to speckle patterns without significant loss of data. These filters make the processing easier for many phase unwrapping methods. This work employed unfiltered images in its comparisons in order to highlight the differences between methods and provide a challenging environment for the swarm, allowing several experiments with the tuning of parameters. In other words, unfiltered images allow the analysis of noisy environments, regardless of the application, since noise might be an inherent feature of certain applications, possibly due to constraints of processing times or the risk of losing data.

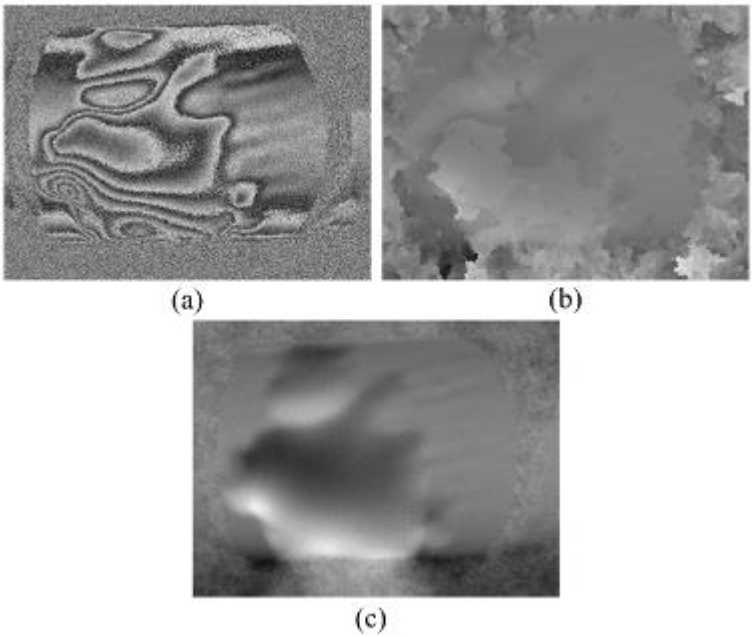
Figure 54a shows a wrapped phase map from a shearography application. Figure 54b shows the phase unwrapped by the quality-based algorithm. It is clear from the resulting phase map that this approach is unable to reliably retrieve the correct phase information from the given image. It is mainly due to the algorithm's nature of lowering its thresholds. Once the threshold is lowered enough to overcome the local noisy boundaries, it will accept unreliable information in future steps. Therefore, the algorithm soon starts to propagate several errors throughout the whole image.

The L^0-norm algorithm, on the other hand, unwraps the phase successfully as shown in Figure 54c at the cost of processing time. This phase map is a good reference to evaluate the swarm-based algorithm's success.

The first experiments with the swarm-based algorithm employed a single threshold at 0.9 and the results are shown in Figure 55. Firstly, the influence of noise over the covered area is noticeable since the same thresholds parameter are unable to cover more than half the image at best.

Secondly, swarm size shows a great influence over the covered area as well, reducing it dramatically with smaller populations. Therefore, it becomes clear that noisy images require more resilient swarms, i.e. swarms that have a faster replication rate and unwrap the phase in higher numbers.

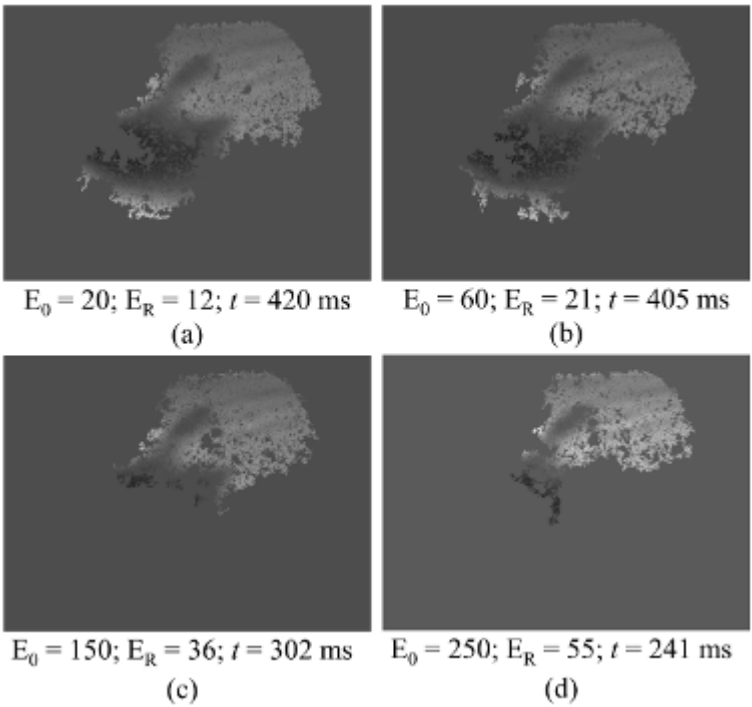
Figure 54 – Fringe pattern (1280x960 pixels) from a shearography application (a) and the unwrapped phase from quality-based (b) and L^0-norm (c) algorithms.



Source: Author's own work.

However, the algorithm's threshold can also be tuned to values that are more appropriate to the current application. Figure 56 shows the results for the same image and energy settings but with a lower threshold, set at 0.8. The influence of the swarm size is even clearer for this configuration, which reinforces the notion that larger populations are more appropriate for noisy applications. The result shown in Figure 56a shows that most of the region with useful information was unwrapped without the propagation of errors similar to those from the quality-based algorithm. This result indicates that the quality evaluation of each agent is indeed reliable, although based on only two pixels at a time.

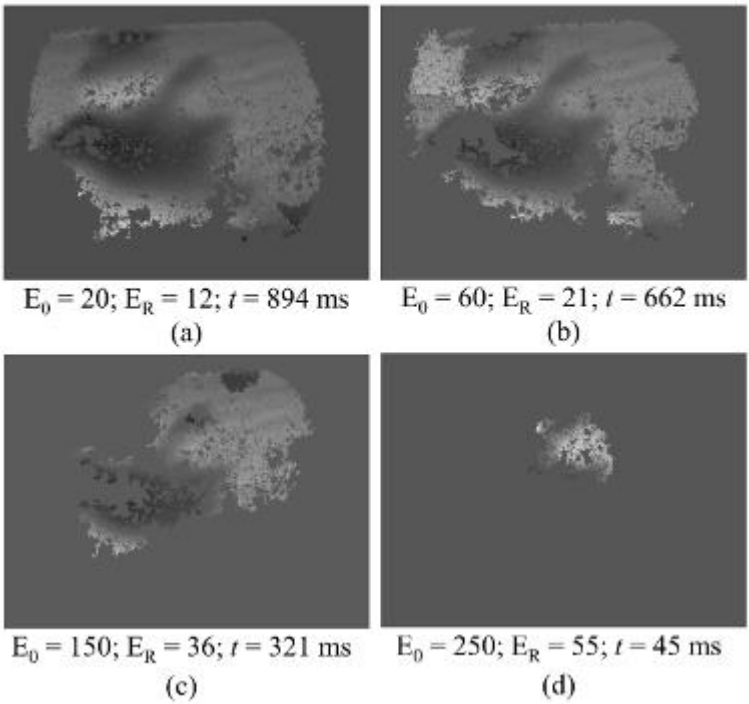
Figure 55 – Results for the swarm-based algorithm with a single threshold at 0.9.



Source: Author's own work.

For some applications, the area covered in Figure 56a would suffice. However, it is important to understand how the swarm could be tuned in order to process further information. The first option is to increase the swarm size by lowering the replication costs. Figure 57a shows that this approach allows the swarm to process the useful region almost completely at the cost of increasing the processing time. The algorithm's reach can be increased by employing more than one threshold.

Figure 56 - Results for the swarm-based algorithm with a single threshold at 0.8.

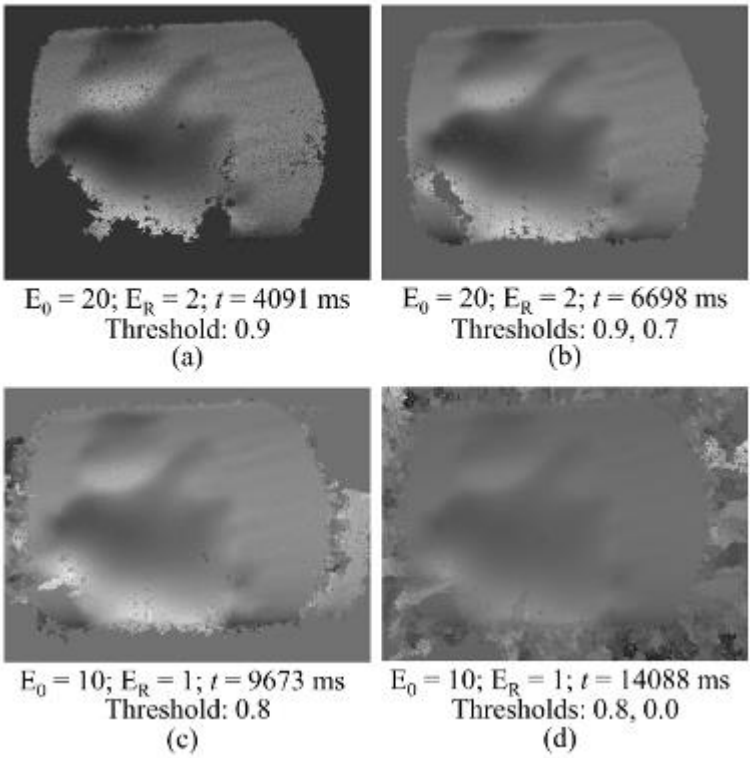


Source: Author's own work.

Multiple threshold are employed by reactivating an agent with a lower threshold after it deactivates for the lack of unwrapping operations. If this agent comes to deactivate again, the threshold is not lowered any further. Figure 57b shows the results from a configuration with two thresholds. The swarm is indeed capable of unwrapping a larger region but increasing the processing times.

These first two results would allow most of the necessary information to be extracted and the overall quality shows how successful the algorithm can be, especially when compared to the tested quality-based algorithm. Even with processing times around a few seconds, this method is still much faster than the expected times required by minimum-norm algorithms as presented in the literature [3, 61].

Figure 57 – Results from different configurations of the swarm-based algorithm.

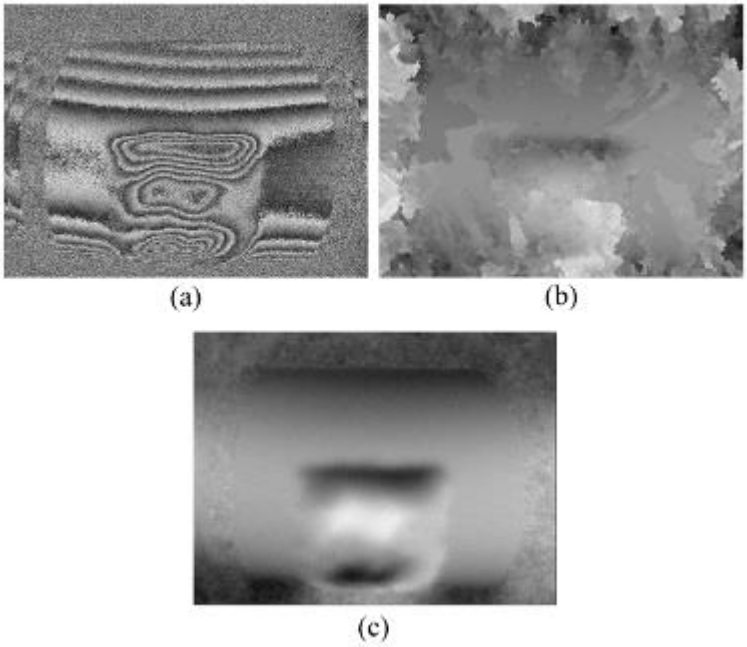


Source: Author's own work.

However, if processing times are not critical to the application and the whole image must be unwrapped, the swarm-based algorithm's reach can be further extended. Figure 57c shows the results for an even larger population, where the agents are able to replicate after each successful unwrap operation. Figure 57d shows this same energy setting employed with an additional zero-level threshold in order for the swarm to process every pixel in the image, independently of its usefulness.

Further experiments were performed with the phase map shown in Figure 58. Similarly to the first example, the quality-based algorithm was unable to process the phase map reliably while the L^0-norm method unwrapped the phase map correctly albeit slowly.

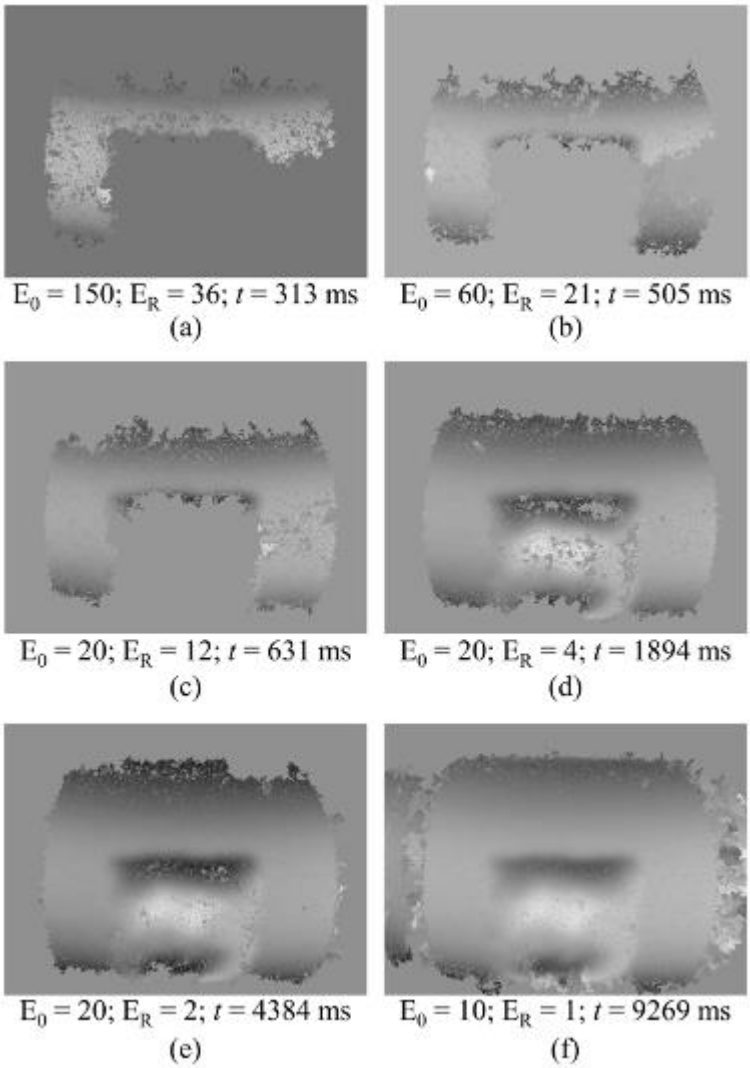
Figure 58 - Fringe pattern (1280x960 pixels) from a shearography application (a) and the unwrapped phase from quality-based (b) and L^0-norm (c) algorithms.



Source: Author's own work.

This phase map shows how the energy settings alone can determine where the algorithm will perform on the quality vs. time scale. Figure 59 shows several results from experiments with an increasing population. The first three results are unable to unwrap a large portion of the phase map and can be considered unsatisfactory for almost any application. The last three configurations, however, are capable of unwrapping the great majority of the useful information and offer satisfactory options for many metrological purposes.

Figure 59 – Results from the swarm-based algorithm employing a single threshold at 0.8.



Source: Author's own work.

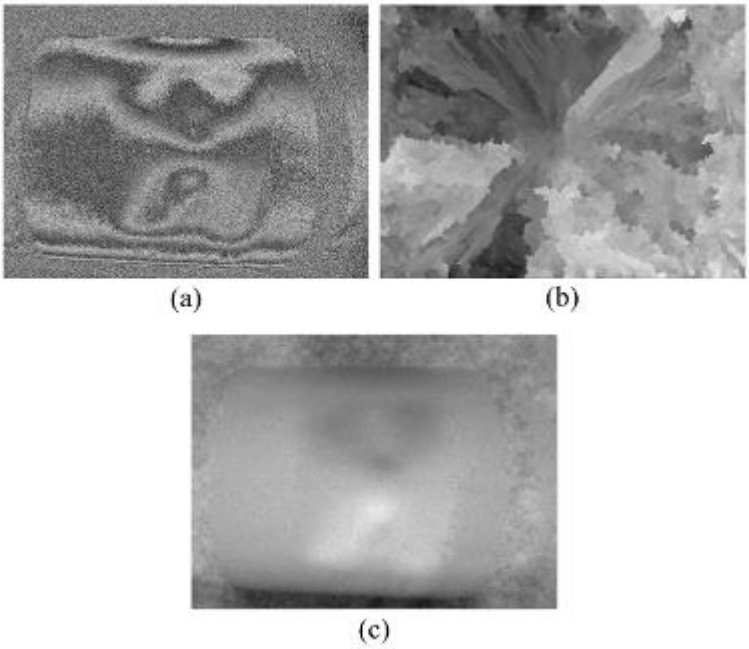
It can be also noted that the processing time increases with an increasing population. On the first experiments, it is mainly due to the covered area that also increases with the population. In other words, the

processing time depended mainly on how many pixels could be processed by the swarm.

On the later experiments, however, the processing time increases dramatically while the covered area is almost the same. It can be concluded that the algorithm becomes less efficient with very dense swarms, probably due to an increase of the unnecessary random searches, and this approach is only appropriate if this little increase on the covered area is fundamental for a specific application.

Finally, an especially difficult phase map, shown in Figure 60a, was also tested and compared. Figure 60b shows the unwrapped phase map from the quality-based algorithm and it can be seen that this method produces even worse results for this noisy fringe pattern. Similarly to the last examples, the L^0-norm method was capable of reliably unwrapping the map despite the high levels of noise, as shown in Figure 60c.

Figure 60 - Fringe pattern (1280x960 pixels) from a shearography application (a) and the unwrapped phase from quality-based (b) and L^0-norm (c) algorithms.



Source: Author's own work.

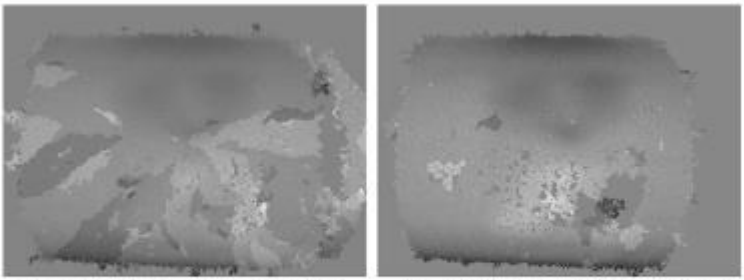
Several experiments were performed with the swarm-based algorithm and it was observed that for any threshold above 0.8, the swarm was unable to unwrap many pixels before it was completely deactivate. However, even if the threshold was set at 0.8 or lower, the swarm was unable to grow if the replication cost was 2 or greater. This means that reliable information was often surrounded by unreliable pixels and the swarm was only able to cover a large area if it was allowed to replicate with every unwrapped pixel it could find.

Therefore, the first configuration able to cover the majority of the useful area was a replication cost of 1 with a threshold set at 0.8. The result is shown in Figure 61a. It can be observed that this configuration allowed the propagation of many errors, compromising the resulting phase map.

A very particular solution was to employ two thresholds: the first slightly above the average quality and the second slightly below. In this manner, the agents always process the good information first, relying on a lower threshold only after all the good pixels were exhausted. These agents than travel over pixels with slightly lower quality and replicate again in regions with better quality, creating agents that again require a higher pixel quality.

The resulting phase map for this configuration is shown in Figure 61b. The increase in the quality of the information over the first configuration is clear. In addition, the processing time was surprisingly lower. This example shows how flexible this approach can be, being tuned for specific applications after a few quick experiments.

Figure 61 – Results from the swarm-based algorithm.



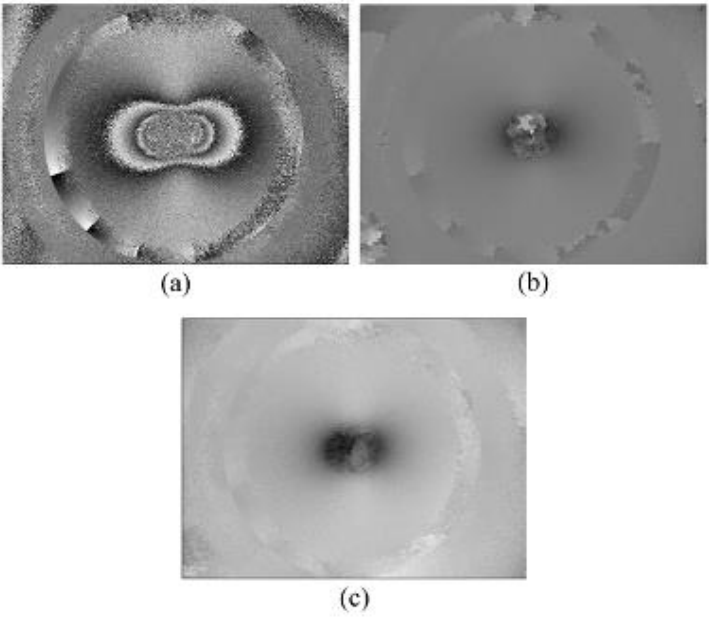
Source: Author's own work.

5.3 CASE STUDY

As previously explained, the main subject for the proposed method is the measurement of residual stress fields through the ESPI technique presented in Section 4.3. This technique generates phase maps similar to those from the shearography application, i.e. containing high levels of noise and regions without information. Therefore, similar results are expected from the swarm-based algorithm. Also similar to the shearography experiments, these images were unfiltered in order to highlight the behavior of the swarm algorithm in comparison to the other methods.

The first experiment was based on the phase map shown in Figure 62a. This map has the region of interest around the machined hole at the center of the image. Despite noisy, the fringes are clear and have a good contrast. Figure 62b shows the phase map unwrapped by the quality-based algorithm. The results are good as are those from the L^0-norm method shown in Figure 62c.

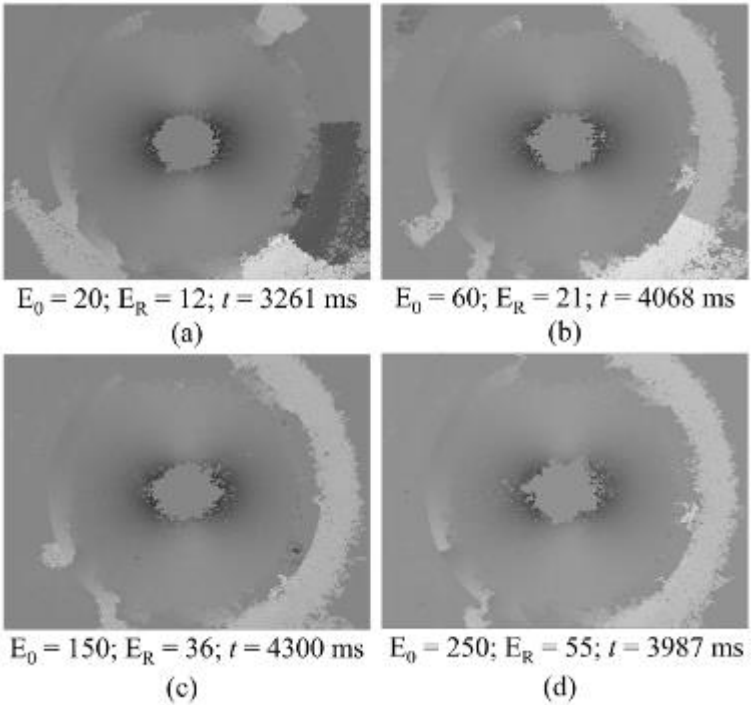
Figure 62 – Wrapped phase map (1600x1200 pixels) from a residual stress measurement (a) and the unwrapped phase map from the quality-based (b) and L^0-norm (c) algorithms.



Source: Author's own work.

Figure 63 shows the results from the swarm-based algorithm. The results are unlike those from the shearography images because the swarm size showed little influence over the covered area. In addition, the region of interest was unwrapped almost completely for all energy settings. Therefore, the fastest configuration, i.e. the larger population, could be employed without compromising the stress measurement.

Figure 63 – Results from the swarm-based algorithm employing a single threshold at 0.9 and the energy presets.

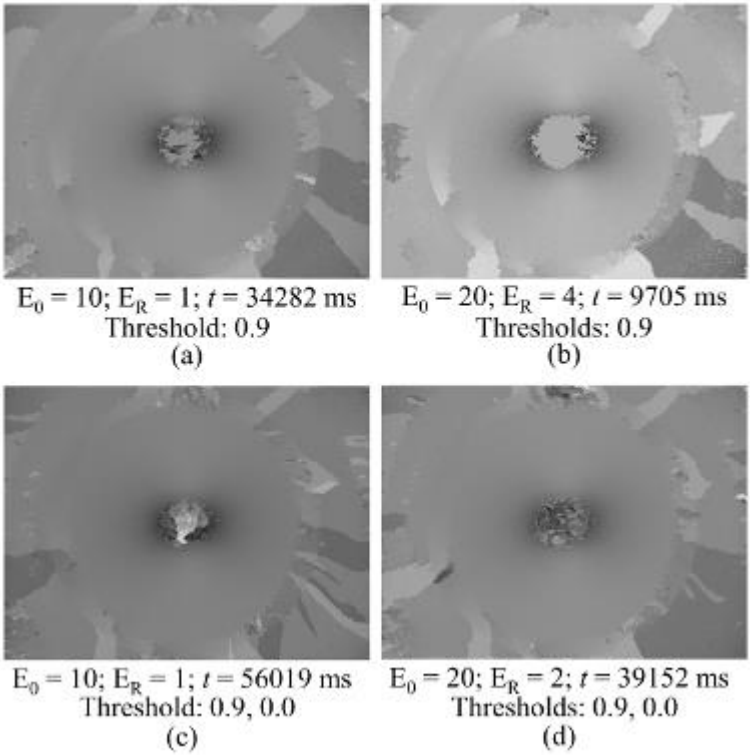


Source: Author's own work.

Nonetheless, it is useful to understand what parameters would produce results similarly to the other tested algorithms. The first option is to increase the swarm size dramatically. The result is shown in Figure 64a. The image is unwrapped almost completely but at a processing cost ten times greater than necessary. A smaller population is able to produce similar results with faster times as shown in Figure 64b. Figure 64c and Figure 64d show even slower results that cover a greater area of the image by employing zero-level thresholds. These options should only be

employed for aesthetic reasons in applications where speed is not critical because most of these regions are useless for the stress measurement application.

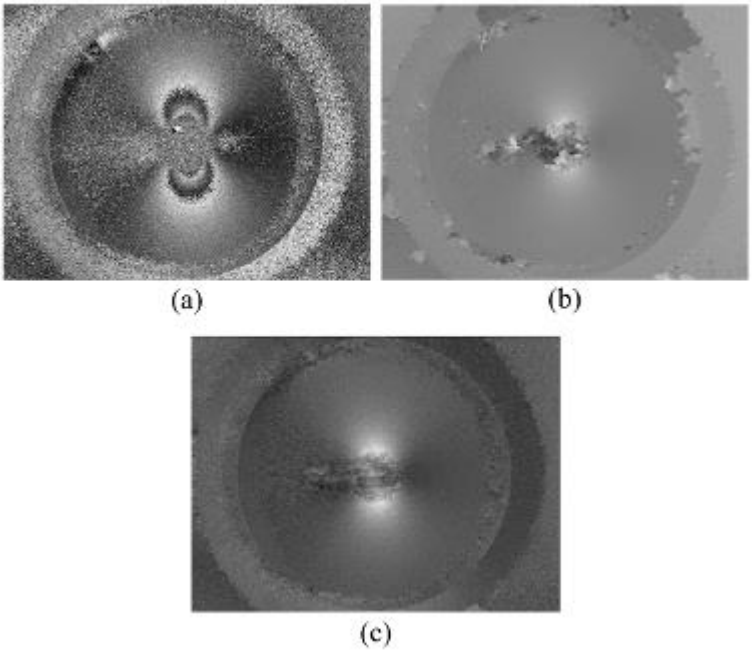
Figure 64 – Results from the swarm-based algorithm using different energy and threshold settings.



Source: Author's own work.

Figure 65 shows another example of an ESPI phase map and the unwrapped solution provided by the other two algorithms. Once again, these algorithms are able to unwrap the phase map successfully. The same configurations employed by the swarm-based algorithm on the first example were tested again.

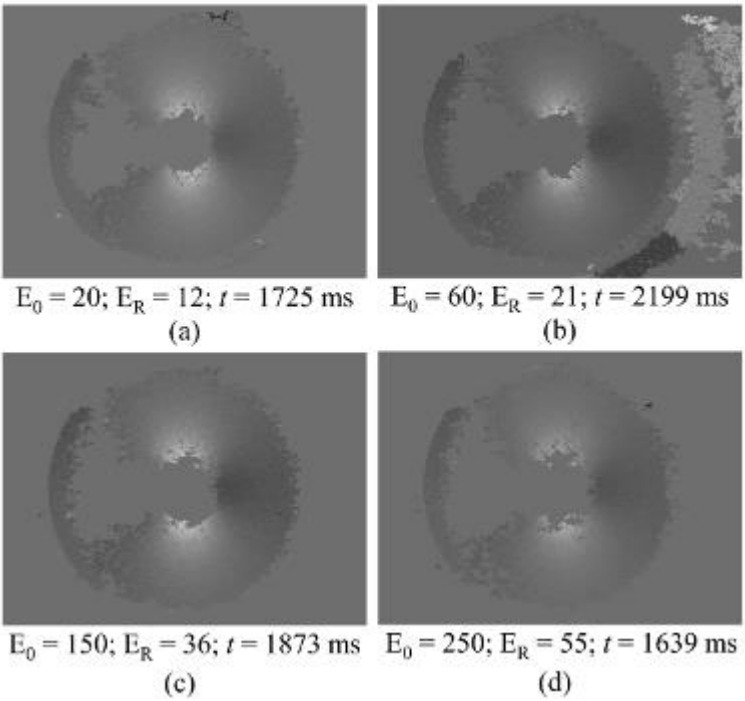
Figure 65 – ESPI wrapped phase map (1600x1200 pixels) (a) and the unwrapped phase maps from the quality-based (b) and L^0-norm (c) algorithms.



Source: Author's own work.

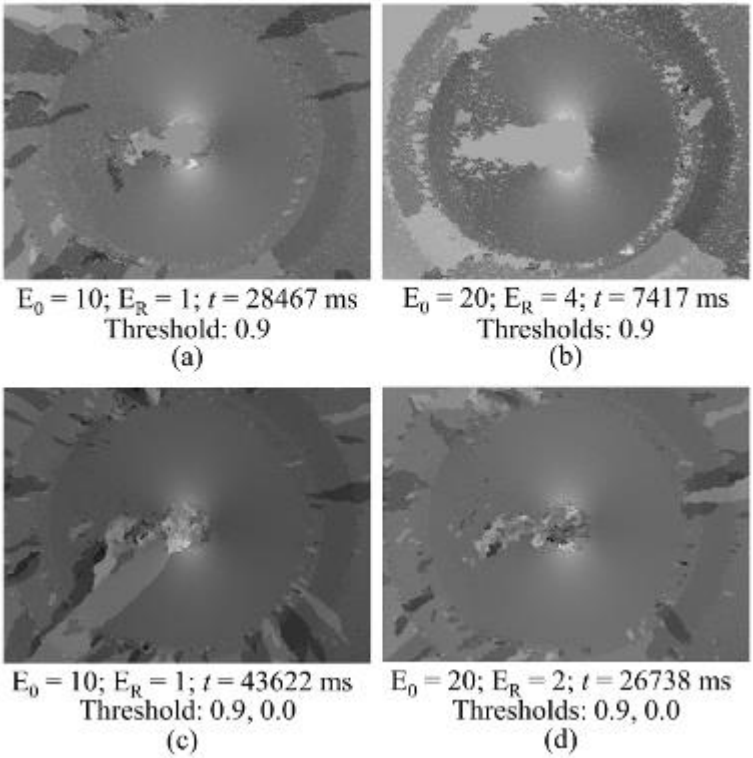
The results presented in Figure 66 show that, again, larger swarms present the best relationship between covered area and processing times. Also similar to the first examples are the results shown in Figure 67. Slightly larger swarms are able to unwrap the great majority of the useful information at the cost of processing speed. The whole image can also be unwrapped by employing zero-level thresholds. However, these settings result in unnecessary processing times and are hardly justifiable.

Figure 66 - Results from the swarm-based algorithm employing a single threshold at 0.9 and the energy presets.



Source: Author's own work.

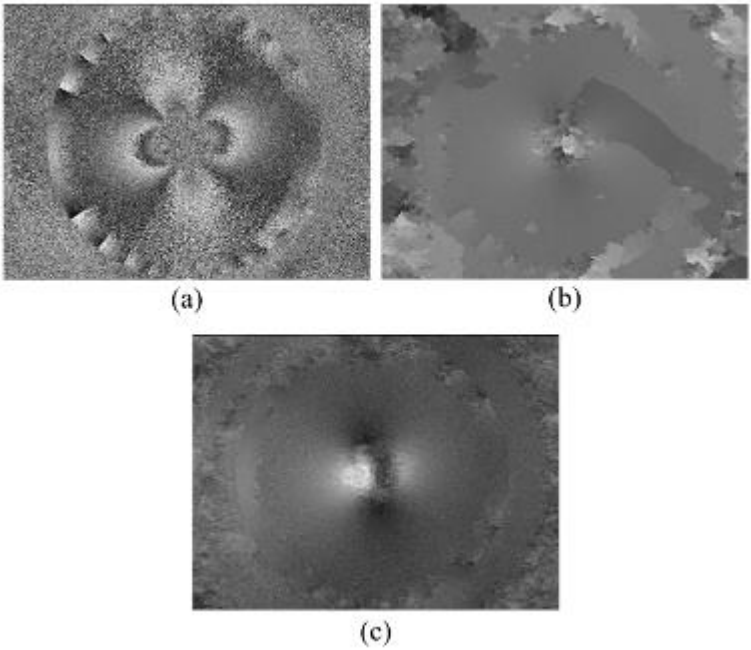
Figure 67 - Results from the swarm-based algorithm using different energy and threshold settings.



Source: Author's own work.

A final example is shown in Figure 68. This phase map contains more noise than the first examples and its impact can be observed on the unwrapped phase map provided by the quality-based algorithm shown in Figure 68b. The noise caused the algorithm to lower its thresholds to levels that propagated errors due to unreliable information. The results from the L^0-norm algorithm were, as usual, of high quality.

Figure 68 - Noisy ESPI wrapped phase map (1600x1200 pixels) (a) and the unwrapped phase maps from the quality-based (b) and L^0-norm (c) algorithms.



Source: Author's own work.

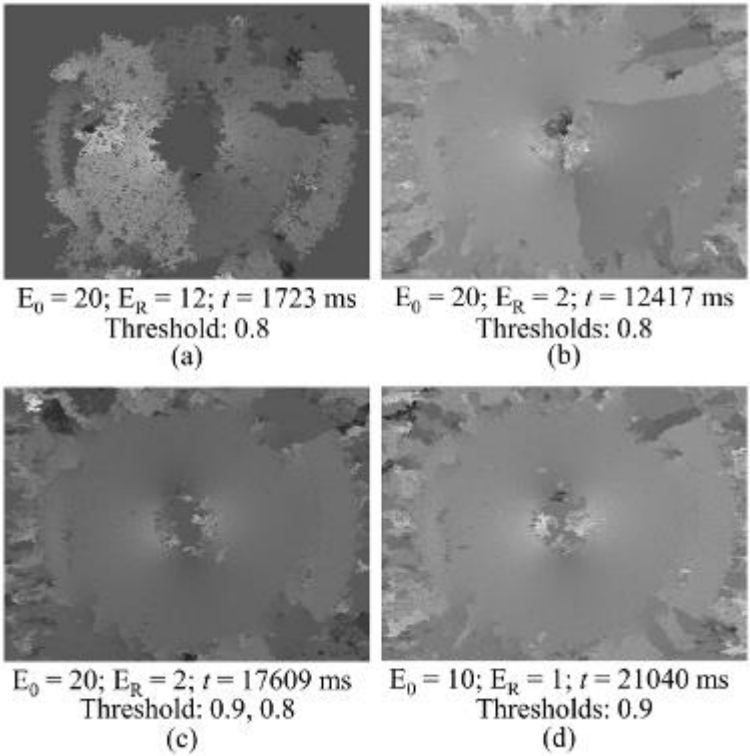
This phase map posed a good challenge for the swarm-based algorithm. As it can be seen in Figure 69a, the usual energy and threshold setting were unable to reliably unwrap the image. Not only was the region of interest sparsely processed, but also many errors were propagated through large portions of the image. Increasing the swarm size enabled the algorithm to cover the image completely. However, a single threshold was unable to provide good results.

A solution was to approach it like the image presented in Figure 61. By setting two thresholds, one above the average pixel quality and another below it, the algorithm was able to reliably unwrap the image almost completely. A similar result was given by a single threshold with much a larger swarm, however slower.

These results show that the proposed method can solve the same problematic phase map through very different approaches and that it depends highly on the application at hand. Notwithstanding, it showed good results when compared to other known algorithms. The algorithm

was able to provide results with a quality similar to those from the L^0-norm method.

Figure 69 - Results from the swarm-based algorithm using different energy and threshold settings.



Source: Author's own work.

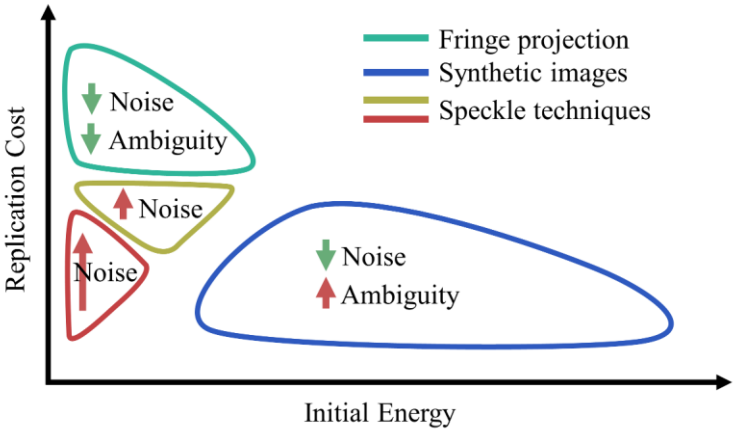
5.4 PARAMETER GUIDELINES

As it was observed in these experiments, the swarm-based solution is extremely flexible. However, it should be tuned to specific applications according to the obstacles featured by each specific case. In order to help with the selection of the parameters, a simple guide was created and it is shown in Figure 70. These rough guidelines aim to provide a qualitative estimation of the energy settings necessary to unwrap a phase map with reliability and fast processing times.

For example, images with low noise levels do not require dense populations. Therefore, agents with higher replication costs can be easily employed. If there are difficult ambiguities to be solved, fewer agents with longer lifespans perform better. If both noise and ambiguities are absent, e.g. as in fringe projection applications, smaller populations with a short lifespan are able to process the image in the fastest time without compromising the results.

For noisy applications, such as speckle-based techniques, the agents must be allowed to replicate at a faster rate. Noisier images require denser and more resilient swarm. In those cases, the use of multiple thresholds is also advisable.

Figure 70 – General guidelines for parameter selection and examples of use.



Source: Author's own work.

6 CONCLUSIONS

This work proposed a novel method to phase unwrapping based on the concept of Swarm Intelligence. Departing from usual SI algorithms based on natural swarms, this work applied the fundamental aspects of emergent behavior to create artificial agents following original rules.

The methodology adopted to develop a working swarm was highly iterative. It was shown that the artificial swarm approach admit more than one solution. The latest solution was selected to be thoroughly tested and compared with existing algorithms. This section will summarize the main results from these experiments.

6.1 PERFORMANCE

The first experiments were based solely on a flawless synthetic image, allowing experimentations with user-defined variables in order to understand the dynamics of the emergent behavior. It was observed that the initial and replication energy settings have a major influence over the swarm density, its capability of reaching unwrappable pixels and the processing times. Larger populations are more thorough on their unwrapping but require longer processing times. Therefore, energy presets were defined in order to balance coverage and processing costs and allow different population sizes for later experiments.

Synthetic images with very little noise but difficult ambiguities were tested. The results confirmed the expected behavior of ambiguity compensations. Unlike the quality-based algorithm tested, the swarm was able to suppress the propagation of errors even after some agents walked over ambiguous paths. It was also shown that smaller populations allowed better suppression of errors.

Images obtained through fringe projection were also tested. The swarm-based algorithm was able to smoothly unwrap the phase map, even better than the quality-based method. However, it shared with the other tested methods the same limitation when dealing with surface discontinuities. Despite this problem, it was shown that smaller swarms did not necessarily unwrap the image faster.

Speckle-based methods were tested, providing images with high levels of noise. Shearography images were tested first and, based on the results from the quality-based algorithm, they posed a great challenge to path-following algorithms. Notwithstanding, the proposed method was able to successfully unwrap the phase maps after proper tuning of the energy and threshold variables.

Images from an ESPI application for the measurement of residual stresses were also experimented. Again, noisy phase maps required more experimentations with the swarm parameters. The proposed method was again capable of reliably unwrapping the phase map, despite requiring longer processing times.

Finally, general guidelines to the selection of parameters were proposed. These guidelines were based on the characteristics of the tested images and the performance results for different energy settings. These guidelines provide a qualitative indication of good parameters to shorten the tests necessary to find the optimal configuration.

6.2 CONTRIBUTIONS

Overall, the results showed that this work's objectives were fulfilled with the successful unwrapped phase maps based on a completely novel approach. The swarm intelligence concept not only proved to be a feasible method for phase unwrapping, but also showed that many solutions can be found and that, perhaps, a better artificial swarm can still be conceived.

When compared to the other path-following algorithm, the swarm-based method showed to be often better in terms of quality of results. Although slower, it was still faster than usual processing times required by minimum-norm algorithms as reported in the literature. Unfortunately, better quality-based algorithms could not be tested and compared. However, this simple comparison is a good evidence of the proposed method's potential.

6.3 LIMITATIONS

The swarm-based method showed some limitations when unwrapping noisy phase maps in terms of processing times. The necessary settings, i.e. dense and resilient swarms, were usually slow to process. Until the method is optimized, there are other methods that claim results as good as those in this work, but faster. The randomness can also be a limitation. For example, for automated applications that rely heavily on reliable phase maps, random propagation of errors could be problematic. Finally, the algorithm must be tuned to specific applications. Even though there are few variables to be adjusted, it is difficult to create one single configuration that reliably unwrap any given phase map.

6.4 SUGGESTIONS FOR FUTURE ADVANCEMENTS

There are many opportunities for the further development of this method. There is certainly the possibility of designing a completely new rule set to perform the phase unwrapping. However, the proposed method already offers many possible advancements.

Firstly, parallel computing could be implemented in order to speed up the process. Theoretically, this technique could achieve processing times comparable to the fastest methods current available, overcoming one of the method's limitations.

In addition, different methods could be used to evaluate the pixel quality. As previously explained, quality parameters are better suited to specific applications. Therefore, other parameters could be tested, perhaps achieving even better results.

Finally, there are still many metrological applications that require phase unwrapping that were left untested. These techniques could provide further challenges to be overcome and grounds to further develop the algorithm's dynamics.

6.5 PUBLICATIONS

This work resulted in two publications during the development of the presented method. The first paper [59] was presented at the SPIE Conference and published at the Conference Proceedings. This paper was based on the first swarm-based solution for phase unwrapping as presented in Section 4.2. The work was well received at the event.

The second paper [60] was published on the same date, coincidentally, at the Applied Optics journal. It focused on the method proposed in Section 4.3, i.e. the rule set that was thoroughly tested in this work.

7 REFERENCES

[1] G. S. SCHAJER, “Hole-drilling residual stress measurement at 75: Origins, advances, opportunities,” *Experimental Mechanics*, vol. 50, pp. 245-253, 2010.

[2] M. R. VIOTTI e A. ALBERTAZZI, “Compact sensor combining DSPI and the hole-drilling technique to measure non-uniform residual stress fields,” in *SPIE Proceedings*, Vigo, 2012.

[3] D. C. GHIGLIA e M. D. PRITT, Two-Dimensional Phase Unwrapping: Theory, Algorithms, and Software, New York: Wiley-Interscience, 1998.

[4] E. BONABEAU, M. DORIGO e G. THERAULAZ, Swarm Intelligence - From Natural to Artificial Systems, New York: Oxford University Press, 1999.

[5] T. YOSHIZAWA, Handbook of Optical Metrology: Principles and Applications, Boca Raton: Taylor & Francis Group, 2008.

[6] E. HETCH, Optics, San Francisco: Addison Wesley, 2002.

[7] K. J. GASVIK, Optical Metrology, Chichester: John Wiley & Sons, 2002.

[8] J. A. LEENDERTZ, “Interferometric displacement measurement on scattering surfaces utilizing speckle effect,” *Journal of Physics E: Scientific Instruments*, vol. 3, pp. 214-218, 1970.

[9] M. VIOTTI e A. ALBERTAZZI JR., “Industrial inspections by speckle interferometry: general requirements and a case study,” in *SPIE Proceedings*, Munich, 2009.

[10] J. A. LEENDERTZ e J. N. BUTTERS, “An image-shearing speckle-pattern interferometer for measuring bending moments,” *Journal of Physics E: Scientific Instruments*, vol. 6, pp. 1107-1110, 1973.

[11] Y. Y. HUNG e C. E. TAYLOR, “Speckle-shearing interferometric camera -- A tool for measurement of derivatives of surface-displacement,” in *SPIE*, 1973.

[12] S. S. GORTHI e R. P., “Fringe Projection Techniques: Whiter we are?,” *Optics and Lasers in Engineering*, vol. 48, nº 2, pp. 133-140, 2010.

[13] K. CREATH, “Phase-measurement interferometry techniques,” in *Progress in Optics XXVI*, Amsterdam, Elsevier, 1988, pp. 349-393.

[14] P. HARIHARAN, B. F. OREB e T. EIJU, “Digital phase-shifting interferometry: a simple error-compensating phase calculation algorithm,” *Applied Optics*, vol. 26, nº 13, pp. 2504-2506, 1987.

[15] P. CARRÉ, “Installation et utilisation du comparateur photoélectrique et interférentiel du Bureau International des Poids et Mesures,” *Metrologia*, vol. 2, nº 1, pp. 13-23, 1966.

[16] M. TAKEDA, H. INA e S. KOBAYASHI, “Fourier-transform method of fringe-pattern analysis for computer-based topography and interferometry,” *Journal of the Optical Society of America*, vol. 72, nº 1, pp. 156-160, 1982.

[17] K. ITOH, “Analysis of the phase unwrapping algorithm,” *Applied Optics*, vol. 21, nº 14, p. 2470, 1982.

[18] R. M. GOLDSTEIN, H. A. ZEBKER e C. L. WERNER, “Satellite radar interferometry: Two-dimensional phase unwrapping,” *Radio Science*, vol. 23, nº 4, pp. 713-720, 1988.

[19] J. M. HUNTLEY, “Noise-immune phase unwrapping algorithm,” *Applied Optics*, vol. 28, nº 15, pp. 3268-3270, 1989.

[20] R. CUSACK, J. M. HUNTLEY e H. T. GOLDREIN, “Improved noise-immune phase-unwrapping algorithm,” *Applied Optics*, vol. 34, nº 5, pp. 781-789, 1995.

[21] J. R. BUCKLAND, J. M. HUNTLEY e S. R. E. TURNER, “Unwrapping noisy phase maps by use of a minimum-cost-matching algorithm,” *Applied Optics*, vol. 34, nº 23, pp. 5100-5108, 1995.

[22] C. W. CHEN e H. A. ZEBKER, “Network approaches to two-dimensional phase unwrapping: intractability and two new algorithms,” *Journal of the Optical Society of America*, vol. 17, nº 3, pp. 401-414, 2000.

[23] Z. SEN, Z. HEPING e T. JINSONG, “Dendriform branch cut algorithm based on minimum spanning tree for phase unwrapping,” *Procedia Engineering*, vol. 29, pp. 1154-1159, 2012.

[24] M. CONSTANTINI, “A novel phase unwrapping method based on network programming,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, nº 3, pp. 813-821, 1998.

[25] S. A. KAROU, M. A. GDEISAT, D. R. BURTON e M. J. LALOR, "Residue vector, an approach to branch-cut placement in phase unwrapping: theoretical study.," *Applied Optics*, vol. 46, nº 21, pp. 4712-4727, 2007.

[26] A. ALBERTAZZI JR, "Robust phase-unwrapping algorithm for discrete meshes," in *Proceedings of SEM Annual Conference*, 1999.

[27] M. ZHAO, L. HUANG, Q. ZHANG, X. SU, A. ASUNDI e Q. KEMAO, "Quality-guided phase unwrapping technique: comparison of quality maps and guiding strategies," *Applied Optics*, vol. 50, nº 33, pp. 6214-6224, 2011.

[28] D. J. BONE, "Fourier fringe analysis: the two-dimensional phase unwrapping problem," *Applied Optics*, vol. 30, nº 25, pp. 3627-3632, 1991.

[29] J. A. QUIROGA, A. GONZÁLEZ-CANO e E. BERNABEU, "Phase-unwrapping algorithm based on an adaptive criterion," *Applied Optics*, vol. 34, nº 14, pp. 2560-2563, 1995.

[30] H. LIM, W. XU e X. HUANG, "Two new practical methods for phase unwrapping," in *Geoscience and Remote Sensing Symposium*, Firenze, 1995.

[31] A. ASUNDI e Z. WENSEN, "Fast phase-unwrapping algorithm based on a gray-scale mask and flood fill," *Applied Optics*, vol. 37, nº 23, pp. 5416-5420, 1998.

[32] X. SU e W. CHEN, "Reliability-guided phase unwrapping algorithm: a review," *Optics and Lasers in Engineering*, vol. 42, pp. 245-261, 2004.

[33] T. J. FLYNN, "Consistent 2-D phase unwrapping guided by a quality map," in *Proceedings of the 1996 International Geoscience and Remote Sensing Symposium*, Piscataway, 1996.

[34] D. C. GHIGLIA e L. A. ROMERO, "Minimum Lp-norm two-dimensional phase unwrapping," *Journal of the Optical Society of America*, vol. 13, nº 10, pp. 1999-2013, 1996.

[35] D. C. GHIGLIA, G. A. MASTIN e L. A. ROMERO, "Cellular-automata method for phase unwrapping," *Journal of the Optical Society of America*, vol. 4, nº 1, pp. 267-280, 1987.

[36] K. A. STETSON, J. WAHID e P. GAUTHIER, "Noise-immune phase unwrapping by use of calculated wrap regions," *Applied Optics*, vol. 36, nº 20, pp. 4830-4838, 1997.

[37] R. YAMAKI e A. HIROSE, “Singularity-spreading,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, nº 10, pp. 3240-3251, 2007.

[38] S. TOMIOKA, S. HESHMAT, N. MIYAMOTO e S. NISHIYAMA, “Phase unwrapping for noisy phase maps using rotational compensator with virtual singular points,” *Applied Optics*, vol. 49, nº 25, pp. 4735-4745, 2010.

[39] J. M. BIOUCAS-DIAS e G. VALADÃO, “Phase unwrapping via graph cuts,” *IEEE Transactions on image processing*, vol. 16, nº 3, pp. 698-709, 2007.

[40] H. WANG, F. LIU e Q. ZHU, “Improvement of phase unwrapping algorithm based on image segmentation and merging,” *Optics Communications*, vol. 308, pp. 218-223, 2013.

[41] S. KIM e Y.-S. KIM, “Two-dimensional phase unwrapping using wavelet transform,” *Electronic Letters*, vol. 38, nº 1, pp. 19-20, 2002.

[42] A. P. ENGELBRECHT, Computational Intelligence - An Introduction, Chichester: John Wiley & Sons, 2007.

[43] C. W. REYNOLDS, “Flocks, herds and schools: a distribuited behavioral model,” *Computer Graphics*, vol. 21, nº 4, pp. 25-34, 1987.

[44] S. GOSS, S. ARON, J. L. DENEUBOURG e J. M. PASTEELS, “Self-organized shortcuts in the Argentine ant,” *Naturwissenschaften*, vol. 76, pp. 579-581, 1989.

[45] U. KIRCHMEIER, S. HAWE e K. DIEPOLD, “A Swarm Intelligence inspired algorithm for countour detection in images,” *Applied Soft Computing*, vol. 13, pp. 3118-3129, 2013.

[46] M. DORIGO, G. DI CARO e L. M. GAMBARDELLA, “Ant algorithms for discrete optimization,” *Artificial Life*, vol. 5, nº 2, pp. 137-172, 1999.

[47] J. KENNEDY e R. EBERHART, “Particle swarm optimization,” em *Proceeding of the IEEE Conference on Neural Networks*, Perth, 1995.

[48] D. TEODOROVIC, P. LUCIC, G. MARKOVIC e M. DELL' ORCO, “Bee Colony Optimization: principles and applications,” in *8th Seminar on Neural Network Applications in Electrical Engineering*, Belgrade, 2006.

[49] D. KARABOGA e B. BASTURK, “A powerful and efficienct algorithm for numerical function optimization: artificial bee colony (ABC) algorithm,” *Journal of Global Optimization,* vol. 39, pp. 549-471, 2007.

[50] M. DORIGO e L. M. GAMBARDELLA, “Ant Colony System: a cooperative learning approach to the Traveling Salesman Problem,” *IEEE Transactions on Evolutionary Computation,* vol. 1, nº 1, pp. 53-66, 1997.

[51] J. E. BELL e S. E. GRIFFIS, “Swarm Intelligence: application of the Ant Colony Optimization algorithm to logistics-oriented vehicle routing problems,” *Journal of Business Logistics,* vol. 31, nº 2, pp. 157-175, 2010.

[52] J. GARCÍA-NIETO, E. ALBA e A. C. OLIVERA, “Swarm intelligence for traffic light scheduling: application to real urban areas,” *Engineering Applications of Artificial Intelligence,* vol. 25, pp. 274-283, 2012.

[53] R. S. KADADEVARAMATH, J. C. H. CHEN, B. L. SHANKAR e K. RAMESHKUMAR, “Application of particle swarm intelligence algorithms in supply chain network architecture optimization,” *Expert Systems with Applications,* vol. 39, pp. 10160-10176, 2012.

[54] S. A. ETEMAD e T. WHITE, “An ant-inspired algorithm for detection of image edge features,” *Applied Soft Computing,* vol. 11, pp. 4883-4893, 2011.

[55] M. SETAYESH, M. ZHANG e M. JOHNSTON, “Edge detection using constrained discrete particle swarm optimisation in noisy images,” in *IEEE Congress on Evolutionary Computation*, New Orleans, 2011.

[56] K. BENATCHA, M. KOUDIL, N. BENKHELAT e Y. BOUKIR, “An algorithm for image segmentation using ants,” in *IEEE International Symposium on Industrial Electronics*, Cambridge, 2008.

[57] X. ZHUANG, G. YANG e H. ZHU, “A model of image feature extraction inspired by ant swarm system,” in *Fourth International Conference on Natural Computation*, Jinan, 2008.

[58] T. MIRZAYANS, N. PARIMI, P. PILARSKI, C. BACKHOUSE, L. WYARD-SCOTT e P. MUSILEK, “A swarm-based system for object recognition,” *Neural Network World,* vol. 15, nº 3, pp. 243-255, 2005.

[59] L. S. MACIEL e A. G. ALBERTAZZI, “Application of a swarm-based approach for phase unwrapping,” in *Proceedings of SPIE, Interferometry XVII: Techniques and Analysis*, San Diego, 2014.

[60] L. S. MACIEL e A. G. ALBERTAZZI, “Swarm-based algorithm for phase unwrapping,” *Applied Optics*, vol. 53, nº 24, pp. 5502-5509, 2014.

[61] E. ZAPPA e G. BUSCA, “Comparison of eight unwrapping algorithms applied to Fourier-transform profilometry,” *Optics and Lasers in Engineering*, vol. 46, pp. 106-116, 2008.

[62] N. S. ROSSINI, M. DASSISTI, K. Y. BENYOUNIS e A. G. OLABI, “Methods of measuring residual stresses in components,” *Materias and Design*, vol. 35, pp. 575-588, 2012.

[63] P. J. WITHERS e H. K. D. H. BHADESHIA, “Residual stress Part 1 - Measurement techniques,” *Materials Science and Technology*, vol. 17, nº 4, pp. 355-365, 2001.

[64] P. J. WITHERS e H. K. D. H. BHADESHIA, “Residual stress Part 2 - Nature and origins,” *Materials Science and Technology*, vol. 17, nº 4, pp. 366-375, 2001.

[65] T. L. F. C. PINTO, *Medição óptica, comparação e sinalização de superfícies com forma livre de grande extensão,* Doctor Thesis - Florianópolis, 2010.